1753 File cy

AD A056347

CONTRACT REPORT ARBRL-CR-00369

TRUCK - A DIGITAL COMPUTER PROGRAM FOR CALCULATING THE RESPONSE OF ARMY VEHICLES TO BLAST WAVES

Prepared by

Kaman AviDyne
Division of Kaman Sciences Corporation
83 Second Avenue
Burlington, Massachusetts

April 1978

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CONTRACT REPORT ARBRL-CR-00369	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TRUCK - A DIGITAL COMPUTER PROGRAM FOR CALCULATING THE RESPONSE OF ARMY VEHICLES TO BLAST WAVES.		5. TYPE OF REPORT & PERIOD COVERED Final Report 19 July 1976 - 4 March 1977
		6. PERFORMING ORG. REPORT NUMBER KA TR-136
7. AUTHOR(s) Norman P. Hobbs, John P. Walsh, Garabed Zartarian, William N. Lee Yau Wu		8. CONTRACT OR GRANT NUMBER(s) DAAD05-74-C-0744
9. PERFORMING ORGANIZATION NAME AND ADDRESS Kaman AviDyne Division of Kaman Sciences Corporation 83 Second Avenue, Burlington, Mass.		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Armament Research & Development Command U.S. Army Ballistic Research Laboratory ATTN: DRDAR-BL Aberdeen Proving Ground, Maryland 21005		12. REPORT DATE APRIL 1978
		13. NUMBER OF PAGES 224
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was sponsored by the U.S. Army Project DA No. 1W162118AH75, AMCMS Code 632306.11.H7200, Nuclear Weapons Effects for Army Applications, Blast and Shock.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Blast Response
Truck Dynamics
Overturning
Computer Simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computer program called TRUCK has been prepared for the purpose of analyzing the response of Army vehicles to blast waves. The program permits the inclusion of non-linear springs and dampers and allows large rotations, thus making it applicable to the overturning problem. The tire-ground interaction model permits the tire to leave the ground, to recontact the ground, to slide along the ground, and to stop sliding. Spring bottoming is handled as an elastic collision. A shelter mounted on the vehicle and equipment racks within the shelter are permitted to undergo motions relative to the vehicle body.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This work was primarily performed for the Ballistic Research Laboratories under Contract Number DAAD05-74-C-0744 by Kaman AviDyne, Burlington, Mass., a division of Kaman Sciences Corporation. Certain improvements to the code, particularly those dealing with the conversion of the code from a response code to a vulnerability code which iterates for critical blast levels, were carried out under Harry Diamond Laboratories Contract Numbers DAAG39-77-M-2557 and DAAG39-77-M-4105. The technical monitor for BRL was Dr. W. Don Allison and the technical monitor for HDL was Mr. Louis Belliveau. Dr. Norman P. Hobbs was the project leader and principal investigator of the study which was performed in the Structural Mechanics Group of Kaman AviDyne headed by Mr. E. S. Criscione.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1

INTRODUCTION

The response of trucks and tracked vehicles to blast waves in a tactical nuclear warfare scenario is of considerable interest to military planners. The present report deals with the development of a computer code, TRUCK, for the prediction of the response of such vehicles to blast waves.

Vehicles of particular interest are those which contain C^3 equipment and may be necessary for the proper functioning of other military elements. A communications shelter mounted on a truck would fall in that category. Such a system may be characterized by a truck body attached to the axles through non-linear springs and dampers, a shelter mounted on the truck body, possibly connected to the truck body by springs, dampers, and pretensioned guy wires, and equipment racks mounted within the shelter on springs and dampers. The TRUCK code is intended to yield the gross motions of the vehicle body and of the several elements mentioned above relative to the vehicle body. Large motions are permitted, so that vehicle overturning may be addressed. Elasticity of the individual elements, for example, the shelter, is not included in the TRUCK code.

Previous work in this area is reported in Reference 1, which considers a single degree of freedom system which rolls about an axis on the ground defined by the tire contact points on the leeward side of a vehicle exposed to a side-on blast wave. The associated computer program is specifically designed to predict vehicle overturning.

The TRUCK code is obviously much more complex than the code described in Reference 1. The TRUCK code includes a large number of degrees of freedom rather than just one. The fact that many degrees of freedom are considered means that associated elements, such as springs/dampers, must be characterized and included in the equations of motion. Bottoming and rebound of the springs are included in the code. The tire/ground interaction forces have to be formulated. In addition, since the TRUCK code will handle blast waves intercepting the vehicle from an arbitrary azimuth angle, the aerodynamic loading representation must be much more extensive than in Reference 1.

Just as the TRUCK code is obviously more complex than the Reference 1 code, it is equally obviously much more cumbersome to use. Data acquisition and preparation are much more difficult, and computer running times are much longer. One option within the TRUCK code permits solution of a relatively simple problem; namely, a three degree of freedom problem involving vehicle roll, heave, and sideslip only. Even this simple subproblem still requires data on the tires necessary to define the tire/ground interaction forces.

---

[1] Ethridge, Noel H., Blast Overturning Model for Ground Targets, BRL Report No. 1889, June, 1976. (AD #B012102L)

In Section 2 of this report, the vehicle model is described. This description includes the definition of the coordinate system, the arrangement of the masses and the springs/dampers, and the modeling of the tire/ground interaction. The aerodynamic loading formulation is described in Section 3, while the equations of motion are presented in Section 4.

The computer program based on the foregoing models is described in Section 5. The solution procedure is presented, the input and output are described, and an example problem is presented.

SECTION 2
VEHICLE MODEL

2.1 Mass Model

The layout of the mass model is sketched in Figure 1. Symmetry about a lengthwise vertical plane is assumed. There are four types of masses involved in the model. The first is the vehicle body itself. It is characterized by its mass, center of gravity position, inertia, and products of inertia. Note that the assumption of a plane of symmetry results in two of the three products of inertia being zero, so that only one product of inertia is needed. For all other types of masses, all products of inertia are assumed to be zero.

The second type of mass is the axle/wheel combination, which is described by its mass, center of gravity location, and its three inertias. Any number of axles may be present (currently restricted to six by computer program dimensions). Also, any two consecutive axles may be combined into a bogie free to rotate about an axis normal to the plane of symmetry.

The shelter is the third type of mass, and a rack is the fourth type. Each is characterized by its mass and inertia properties. The racks are assumed to be symmetrically located on the left and right sides of the vehicle. Any number of racks may be present, although current computer program dimensions restrict the number to about two or three on each side of the vehicle, the permissible number depending upon the remainder of the model configuration.

The program may be run with the shelter mass combined with the vehicle mass, with the rack masses combined with the shelter mass, or with the rack and shelter masses combined with the vehicle mass. Another option provides for running with only one mass having only roll, heave, and sideslip degrees of freedom.

2.2 Coordinate Systems

The primary coordinate system is a body axis system which is attached to the vehicle body with its origin at the overall system center of gravity. It is assumed that deformations of the vehicle system (e.g., rack motion relative to the vehicle body) are small in comparison with typical system dimensions. Hence, the center of gravity remains located at the same point on the vehicle body and the mass matrix, involving masses, static unbalances, and inertias, remains constant.

The x, y, and z directions, shown in Figure 1, form an orthogonal right handed body axis system. The right handed rotations about these body axes, ϕ, θ, and ψ, respectively, will be referred to as pitch, yaw, and roll. Translational motions in the x, y, and z directions will be referred to as sideslip, heave, and fore-and-aft translation, respectively.

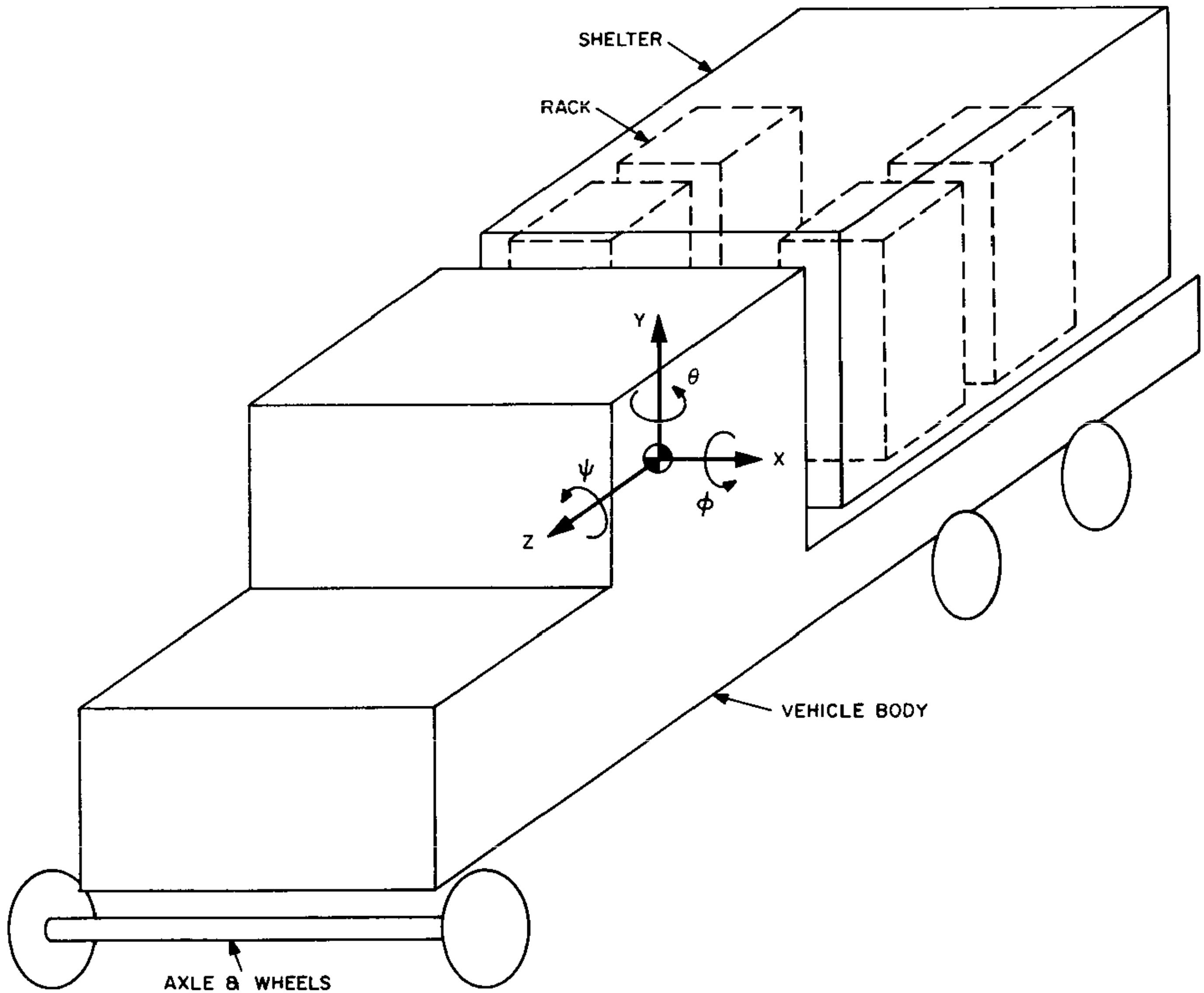

FIGURE 1. MASS MODEL LAYOUT

In addition to the body-axis system just defined, the motions of the axles, shelter, and racks <u>relative</u> to the truck body are described by the degrees of freedom listed in Table 1. Component rotations are about axes through the center of gravity of the individual component. In Table 1, the letter "g" is used to indicate a generalized coordinate, the subscript indicates the direction of the degree of freedom, and the superscript indicates the mass type involved. The order of the degrees of freedom is as indicated in Table 1; that is, the overall vehicle degrees of freedom, the axle degrees of freedom, the shelter degrees of freedom, and the rack degrees of freedom. The axles are presumed to be consecutive from front to back. The left side racks will all precede their right side counterparts in program storage and output. For each mass type, the order of the degrees of freedom is roll, sideslip, heave, pitch, yaw, and fore-and-aft translation. For the axles and the shelter, not all of these degrees of freedom are allowed; the order remains intact, however. The total number of degrees of freedom is seen to be 10 + 2a + 12r, where "a" is the number of axles and "r" is the number of left side racks. If the shelter is taken as rigidly attached to the vehicle body, the total number of degrees of freedom is 6 + 2a + 12r. The dimensions of the current program restrict the number of degrees of freedom to 50 or less.

The first six degrees of freedom will always be the overall vehicle degrees of freedom. It will be convenient in some cases to refer to them as $g_1 \longrightarrow g_6$, as indicated in Table 1.

Two configurations of interest require different interpretations of the degrees of freedom. The first is a tracked vehicle. On a tracked vehicle, there is no axle connecting corresponding tires on either side of the vehicle. Rather, each tire is independently suspended, each with its own freedom in the y direction. If, however, two corresponding tires are assumed to be connected by an imaginary axle, the roll and heave of the axle are sufficient to reproduce the two heave motions of the individual tires. One way of looking at it is to view the two degrees of "axle" freedom as actually being (1) the average of the two tire heave motions, and (2) proportional to the difference between the heave motions. Viewed in this way, the use of axle roll and heave is equivalent to redefining the generalized coordinates to be used to describe the heave motions of the two tires.

The second configuration which requires further discussion is the pitch bogie. Two consecutive axles may be combined into a bogie which is free to pitch about an axis midway between the two axles. The available degrees of freedom, four for the two axles, are adequate to describe the motions. All that is necessary is to add the constraint that the pitching moment about the pitch axis must be zero. This results simply in the bogie suspension spring forces being equally divided between the two axles involved.

TABLE 1

VEHICLE MODEL DEGREES OF FREEDOM

DEGREE OF FREEDOM	MASS TYPE	DESCRIPTION
g_{ψ}^{V} (g_1)	Vehicle	Vehicle roll
g_{x}^{V} (g_2)		Vehicle sideslip
g_{y}^{V} (g_3)		Vehicle heave
g_{ϕ}^{V} (g_4)		Vehicle pitch
g_{θ}^{V} (g_5)		Vehicle yaw
g_{z}^{V} (g_6)		Vehicle fore-and-aft translation
$g_{\psi}^{A_n}$	Axle n	Axle n roll relative to vehicle body
$g_{y}^{A_n}$		Axle n heave relative to vehicle body
g_{ψ}^{S}	Shelter	Shelter roll relative to vehicle body
g_{y}^{S}		Shelter heave relative to vehicle body
g_{ϕ}^{S}		Shelter pitch relative to vehicle body
g_{z}^{S}		Shelter fore-and-aft translation relative to vehicle body
$g_{\psi}^{R_m}$	Rack m	Rack m roll relative to vehicle body
$g_{x}^{R_m}$		Rack m sideslip relative to vehicle body
$g_{y}^{R_m}$		Rack m heave relative to vehicle body
$g_{\phi}^{R_m}$		Rack m pitch relative to vehicle body
$g_{\theta}^{R_m}$		Rack m yaw relative to vehicle body
$g_{z}^{R_m}$		Rack m fore-and-aft translation relative to vehicle body

Since large motions of the vehicle are to be considered, it is clear that $g_1 \longrightarrow g_6$ do not define the location and orientation of the vehicle body. This is true because these motions take place along and about constantly changing directions. In order to determine tire-ground interactions and in order to calculate the blast loads, it is necessary to know the location and orientation of the vehicle. For this reason, an earth-fixed axis system and Euler angles are introduced. These will be denoted by $\bar{g}_1 \longrightarrow \bar{g}_6$, where $\bar{g}_1$ is the Euler roll angle, $\bar{g}_2$ is the motion of the center of gravity in the earth-fixed $\bar{x}$ direction, and so forth. The earth-fixed $\bar{x}$ and $\bar{z}$ axes are horizontal and the $\bar{y}$ axis is vertical. The origin of the earth-fixed axis system is defined as follows. Consider the vehicle with the x-z body axes horizontal. Place the vehicle such that the tires just contact the ground (assumed to be horizontal for the moment). The origin of the earth-fixed axis system is then at the vehicle center of gravity, and the earth-fixed and body axis systems are coincident with the vehicle so placed. Note that the $\bar{x}$, $\bar{y}$, $\bar{z}$ values at blast arrival will not be zero, because the vehicle must settle from the specified position under the action of gravity.

The present formulation permits the ground to slope in the $\bar{x}$ direction; that is, the slope is restricted such that all ground points at the same value of $\bar{x}$ will have the same value of $\bar{y}$. The origin of the earth-fixed axis system with the vehicle on a slope is the center of gravity of the vehicle with the vehicle placed such that the tires just contact an imaginary horizontal ground plane at the same height as the actual ground directly below the center of gravity.

The Euler angular rotations are taken in the conventional order $\bar{\psi}$, $\bar{\theta}$, $\bar{\phi}$ (which are the same as $\bar{g}_1$, $\bar{g}_5$, $\bar{g}_4$). The Euler angular velocities are found from the body-axis angular velocities from the relations (see, for example, Reference 2)

$$\dot{\bar{\psi}} = \frac{1}{\cos\bar{\theta}} (\dot{g}_1 \cos\bar{\phi} + \dot{g}_5 \sin\bar{\phi}) \tag{1a}$$

$$\dot{\bar{\theta}} = \dot{g}_5 \cos\bar{\phi} - \dot{g}_1 \sin\bar{\phi} \tag{1b}$$

$$\dot{\bar{\phi}} = \dot{g}_4 + \dot{\bar{\psi}} \sin\bar{\theta} \tag{1c}$$

Here a dot indicates differentiation with respect to time. The Euler angles are then found by integrating the angular velocities.

Translational velocities in the earth-fixed and body axis systems may be related through a transformation matrix, β.

---

[2] Etkin, Bernard, "Dynamics of Flight", John Wiley and Sons, Inc., New York, 1959.

$$\begin{Bmatrix} \dot{\bar{g}}_2 \\ \dot{\bar{g}}_3 \\ \dot{\bar{g}}_6 \end{Bmatrix} = \begin{bmatrix} & & \\ & \beta & \\ & & \end{bmatrix} \begin{Bmatrix} \dot{g}_2 \\ \dot{g}_3 \\ \dot{g}_6 \end{Bmatrix} \tag{2}$$

where

$$\beta = \begin{bmatrix} (\cos\bar{\theta}\ \cos\bar{\psi}) & \begin{pmatrix} -\cos\bar{\phi}\ \sin\bar{\psi} \\ +\sin\bar{\phi}\ \sin\bar{\theta}\ \cos\bar{\psi} \end{pmatrix} & \begin{pmatrix} \sin\bar{\phi}\ \sin\bar{\psi} \\ +\cos\bar{\phi}\ \sin\bar{\theta}\ \cos\bar{\psi} \end{pmatrix} \\ (\cos\bar{\theta}\ \sin\bar{\psi}) & \begin{pmatrix} \cos\bar{\phi}\ \cos\bar{\psi} \\ +\sin\bar{\phi}\ \sin\bar{\theta}\ \sin\bar{\psi} \end{pmatrix} & \begin{pmatrix} -\sin\bar{\phi}\ \cos\bar{\psi} \\ +\cos\bar{\phi}\ \sin\bar{\theta}\ \sin\bar{\psi} \end{pmatrix} \\ (-\sin\bar{\theta}) & (\sin\bar{\phi}\ \cos\bar{\theta}) & (\cos\bar{\phi}\ \cos\bar{\theta}) \end{bmatrix} \tag{3}$$

The location of the vehicle center of gravity is found by integrating the earth-fixed axis system velocities.

2.3 Representation of Springs and Dampers

The various masses described previously are connected by springs and dampers. The springs and dampers are characterized by their mechanical properties and by their attachment points to the masses they interconnect.

The mechanical properties required are simply the curves of force vs displacement for springs and force vs velocity for dampers. The conventions used herein are as follows:

Springs:

Shortening is a negative displacement and gives a positive force.

Elongation is a positive displacement and gives a negative force.

Dampers:

The compression stroke corresponds to a negative velocity and a positive force.

The rebound stroke corresponds to a positive velocity and a negative force.

Each spring or damper curve is approximated by a series of straight lines, so that non-linear behavior is accounted for. No hysteresis effects are included, however; thus, loading and unloading take place along the same curve.

Consider a spring connecting mass "r" to mass "s". The coordinates at which the spring attaches to mass r will be designated by ℓ_{x_r}, ℓ_{y_r}, and ℓ_{z_r}, where the lengths, ℓ, are measured from the center of gravity of mass r, and similarly for mass s. The elongation of spring "i", δ_i, may be found from the generalized displacements, g_j.

$$\delta_i = \lfloor \lambda_{ij} \rfloor \quad \{g_j\} \tag{4}$$

where λ_{ij} will be zero except for j corresponding to degrees of freedom of mass r or mass s. It should be noted that $g_1 \longrightarrow g_6$ are never involved in spring elongations, since they refer to overall motions of the entire vehicle. Hence, the j dimension of the matrix λ_{ij} can be 6 smaller than the total number of degrees of freedom. Note also that the same λ matrix transforms generalized velocities, $\dot{g}_j$, into elongational velocities, $\dot{\delta}_i$.

The spring direction cosines are

$$a \equiv (L_{x_s} - L_{x_r})/L \tag{5a}$$

$$b \equiv (L_{y_s} - L_{y_r})/L \tag{5b}$$

$$c \equiv (L_{z_s} - L_{z_r})/L \tag{5c}$$

where the subscripted L's are measured from <u>any</u> common reference, and

$$L \equiv \sqrt{(L_{x_s}-L_{x_r})^2+(L_{y_s}-L_{y_r})^2+(L_{z_s}-L_{z_r})^2} \tag{6}$$

The spring elongation is then given by

$$\delta = [g_{x_s} - g_{x_r} + \ell_{z_s} g_{\theta_s} - \ell_{z_r} g_{\theta_r} - \ell_{y_s} g_{\psi_s} + \ell_{y_r} g_{\psi_r}]a$$

$$+ [g_{y_s} - g_{y_r} - \ell_{z_s} g_{\phi_s} + \ell_{z_r} g_{\phi_r} + \ell_{x_s} g_{\psi_s} - \ell_{x_r} g_{\psi_r}]b \qquad (7)$$

$$+ [g_{z_s} - g_{z_r} + \ell_{y_s} g_{\phi_s} - \ell_{y_r} g_{\phi_r} - \ell_{x_s} g_{\theta_s} + \ell_{x_r} g_{\theta_r}]c$$

The elements of the λ matrix may be easily deduced by comparing Eqs. 4 and 7.

The present formulation permits guy wires to attach the shelter to the vehicle body. These are handled exactly as ordinary springs except that a guy wire preload may be specified. That is, the value of the tensile load in a guy wire with the truck in a trimmed condition under the action of gravity may be specified by the analyst. All other springs will end up with whatever load is consistent with a trimmed condition.

Spring bottoming is allowed. When a spring bottoms out, an elastic collision conserving kinetic energy is assumed. The collision process involves an instantaneous change in velocities, effectively due to the application of an impulse at the collision point. If the velocity of a point is denoted by v_x, v_y, and v_z and the change in velocity due to the collision by Δv_x, Δv_y, and Δv_z, conservation of kinetic energy means that

$$\int \left\{ (v_x+\Delta v_x)^2+(v_y+\Delta v_y)^2+(v_z+\Delta v_z)^2- v_x^2-v_y^2-v_z^2 \right\} dm = 0 \qquad (8)$$

where dm is an elemental mass and the integral extends over all masses. Equation 8 may be rewritten

$$2\int \left\{ v_x\Delta v_x+v_y\Delta v_y+v_z\Delta v_z \right\} dm + \int \left\{ \Delta v_x^2+\Delta v_y^2+\Delta v_z^2 \right\} dm = 0 \qquad (9)$$

If a unit impulse applied at the collision point produces velocity changes δv_x, δv_y, and δv_z, then

$$\Delta v_x = I\delta v_x \tag{10}$$

and so forth, where I is the value of the applied impulse. Substituting Eq. 10 into Eq. 9 and rearranging,

$$I = -2\frac{\int\left\{v_x\delta v_x+v_y\delta v_y+v_z\delta v_z\right\}dm}{\int\left\{\delta v_x^{\,2}+\delta v_y^{\,2}+\delta v_z^{\,2}\right\}dm} \tag{11}$$

The integrals in the above equation may be expressed in terms of the present mass model as

$$\int\left\{v_x\delta v_x+v_y\delta v_y+v_z\delta v_z\right\} dm = [\dot{g}_i]\ [M_{i,j}]\ \left\{\delta\dot{g}_j\right\} \tag{12}$$

$$\int\left\{\delta v_x^{\,2}+\delta v_y^{\,2}+\delta v_z^{\,2}\right\} dm = [\delta\dot{g}_i]\ [M_{i,j}]\ \left\{\delta\dot{g}_j\right\} \tag{13}$$

where $\delta\dot{g}$ is the generalized velocity change due to the application of a unit impulse at the collision point and $M_{i,j}$ is the generalized mass matrix, defined in Division 4.2, Table 3.

The velocities, $\dot{g}$, are known at the time of collision. The velocity changes due to a unit impulse, $\delta\dot{g}$, can easily be found by applying the equations of motion developed in Section 4. Equations 12 and 13 then allow the integrals to be evaluated, and Eq. 11 may then be used to determine the value of the required impulse. The velocity changes may then be found from Eq. 10. Finally, the new velocities are found by adding the velocity changes to the velocities which existed just prior to the collision.

2.4 Tire-Ground Interaction Forces

The tire-ground interaction is based upon a Coulomb friction representation. The force on the tire consists of a component normal

to the ground surface and a component tangential to the ground surface. These will be referred to simply as the normal and tangential forces. The normal force will depend upon the normal deflection and normal velocity of the tire and the tire spring and damping characteristics. The tangential force representation separates into two regimes, non-sliding and sliding. While the tire is not sliding, the tangential force depends upon the tire tangential deflection and velocity and the spring and damping characteristics of the tire. When the tangential spring force reaches a limit value, equal to the tire normal force times the coefficient of friction, the tire starts to slide. The tangential tire force during sliding is taken as equal to the normal force times the coefficient of friction and is in a direction opposite to the tangential deflection.

A tire is also allowed to leave the ground, at which time the ground-interaction forces become zero, of course. When the tire contacts the ground again, the tangential tire deflection is set equal to zero.

The mathematical implementation of the above concepts turns out to be rather complex. A brief description of the development is given below.

The ground geometry is shown in Figure 2. The ground surface is assumed to consist of two intersecting planes, with the intersection line parallel to the earth-fixed $\bar{z}$ axis. The ground slopes in the two regions are γ_1 and γ_2, and the slope change occurs at $\bar{x} = x_{sc}$.

The position of the center of the tire in the earth-fixed axis system is given by

$$\begin{Bmatrix} \bar{x}_t \\ \bar{y}_t \\ \bar{z}_t \end{Bmatrix} = \begin{bmatrix} \beta \end{bmatrix} \begin{Bmatrix} L_x - \ell_y g_\psi \\ L_y + g_y + \ell_x g_\psi \\ L_z \end{Bmatrix} + \begin{Bmatrix} \bar{g}_2 \\ \bar{g}_3 + R - L_y \\ \bar{g}_6 \end{Bmatrix} \tag{14}$$

In Eq. 14, the ℓ's are measured from the axle center of gravity to the center of the tire and the L's are measured from the vehicle center of gravity to the center of the tire. The generalized displacements g_y and g_ψ are for the axle associated with the tire being considered. The $\bar{y}_t$ distance is biased such that, with the vehicle wheels just touching the ground, $\bar{y}_t$ will be equal to R, the tire radius. This may be verified by observing that $\bar{g}_3$, g_y, and g_ψ are all zero for the pre-blast condition which establishes the origin for the earth-fixed axis system.

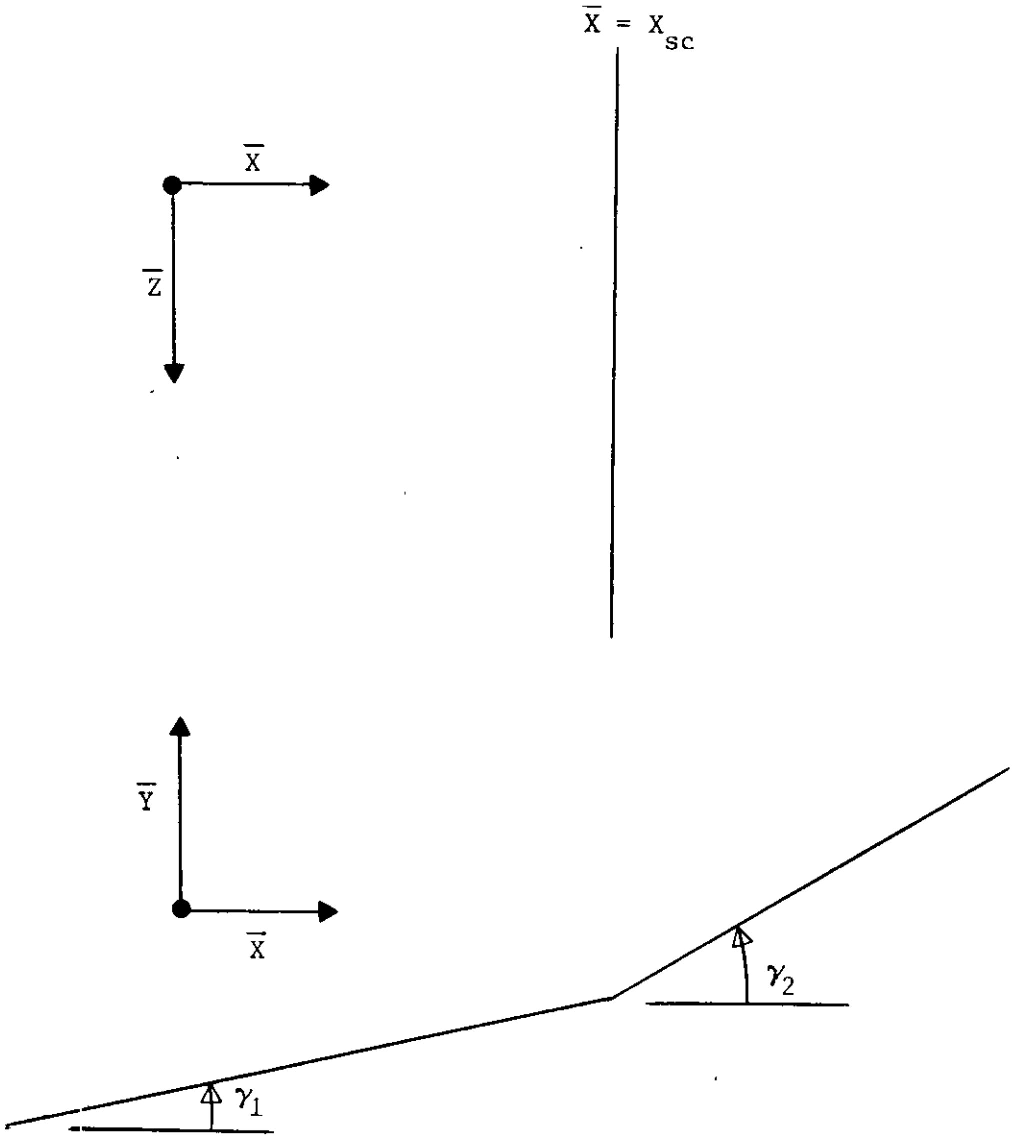

FIGURE 2. GROUND GEOMETRY

Knowing the position of the center of the tire and knowing the ground surface configuration, the height of the center of the tire above the ground, measured in the plane of the tire, may be determined. The expression for the tire height, which is the result of a moderately lengthy development, is

$$H = \frac{[\bar{y}_t - \bar{x}_t \tan\gamma - A]\, \ell_a}{\sqrt{\ell_a^2\,(1+\tan^2\gamma)-(\eta-\xi\tan\gamma)^2}} \tag{15}$$

where

$$\eta \equiv \bar{y}_t\ (\text{left tire})-\bar{y}_t(\text{right tire}) \tag{16}$$

$$\xi \equiv \bar{x}_t\ (\text{left tire})-\bar{x}_t(\text{right tire}) \tag{17}$$

and ℓ_a is the axle length. The parameter A characterizes the ground configuration, and is defined in region 1 by

$$A = 0 \text{ if } x_{sc} \geq 0$$

$$A = x_{sc}\ (\tan\gamma_2-\tan\gamma_1) \text{ if } x_{sc} < 0 \tag{18a}$$

and in region 2 by

$$A = -\, x_{sc}\ (\tan\gamma_2-\tan\gamma_1) \text{ if } x_{sc} \geq 0$$

$$A = 0 \text{ if } x_{sc} < 0 \tag{18b}$$

Equation 15 is used for each of the two regions, and the smaller value of H is used if the ground contour is concave, and the larger is used if the ground contour is convex. The corresponding region is noted, and the slope of that region is used in ensuing calculations. It may easily be verified that in the case of zero ground slope with the axle horizontal (η=0), Equation 15 gives $H = \bar{y}_t$, as it should.

The tire in-plane deflection is

$$\delta = H - R \tag{19}$$

If the deflection is negative, the force is found from the tire characteristics. The component of this force normal to the ground is then used. If the deflection is positive, the tire is off the ground and all tire forces are zero.

Assuming the tire to be rigid, the body-axis components of the velocity of the tire contact point are given by

$$\left\{V_t\right\} = \left\{\begin{matrix} -(L_y-R)\dot{g}_1 + \dot{g}_2 + L_z\dot{g}_5 - (\ell_y-R)\dot{g}_\psi \\ L_x\dot{g}_1 + \dot{g}_3 - L_z\dot{g}_4 + \ell_x\dot{g}_\psi + \dot{g}_y \\ (L_y-R)\dot{g}_4 - \ell_x\dot{g}_5 + \dot{g}_6 \end{matrix}\right\} \tag{20}$$

This velocity must be broken down into components normal to and tangential to the ground. This is done by using a unit vector normal to the ground, N, the body axis components of which are given by

$$\left\{N\right\} = \left[\beta\right]^T \left\{\begin{matrix} -\sin\gamma \\ \cos\gamma \\ 0 \end{matrix}\right\} \tag{21}$$

where it should be noted that the transpose of β is equal to the inverse of β.

The velocity normal to the ground is simply the dot product of the V_t and the N vectors, and the associated components are

$$\left\{V_N\right\} = \left([N] \left\{V_t\right\} \right) \left\{N\right\} \tag{22}$$

The term in parentheses is a scalar and is simply the magnitude of the normal velocity, except that it may be either positive or negative.

Knowing the total velocity and the velocity normal to the ground, the tangential velocity, V_T, is simply the difference.

$$\left\{V_T\right\} = \left\{V_t\right\} - \left\{V_N\right\} \tag{23}$$

The normal and tangential velocities are used to calculate tire damping forces, and the tangential velocity is integrated to determine the tangential displacement, δ_T, which is in turn used to find the tire tangential spring force. The tangential velocities of the front tires (axle 1) are given special treatment, however, on the assumption that

the front tires are free to roll in their planes. Therefore, for a front tire, the component of the tangential velocity which is normal to the intersection of the wheel plane and the ground plane is used rather than the total tangential velocity.

If the tire is in contact with the ground and is not sliding, the normal and tangential spring and damping forces are found as indicated above. The spring forces are applied in directions opposite to the displacements, and the damping forces are applied in directions opposite to the velocities. The maximum allowable tangential force, μF_N^δ, where μ is the coefficient of friction and F_N^δ is the normal spring force, is continuously monitored. When the tangential deflection becomes such that the corresponding spring force, F_T^δ, is greater than μF_N^δ, the tire is allowed to slide. The tangential damping force, $F_T^{\dot{\delta}}$, is set equal to zero and F_T^δ is set equal to μF_N^δ and is taken to act in opposition to the tangential displacement.

During sliding, the magnitude of the tangential deflection is always equal to the value corresponding to the force μF_N^δ. The direction of the tangential deflection is found as follows. First of all, a more specific definition of "tangential deflection" is required. To do so, two tire-ground contact points are defined. The first is the contact point which would exist if the tire were rigid, which will be referred to as the rigid-tire contact point. The velocities given by Eq. 20 are the velocities of this point. The second contact point is the actual contact point between the tire and the ground. The tangential deflection is the vector from the actual contact point to the rigid-tire contact point.

During each time step, the motion of the actual contact point is assumed to be in the direction of the previous tangential deflection. The rigid-tire contact point moves in the direction of the tangential velocity an amount equal to that velocity times the integration time step. The position of the rigid-tire contact point is thus known, the magnitude of the tangential deflection is known, and the line along which the actual contact point must lie is known. These facts are sufficient to permit definition of the location of the actual contact point.

When the movement of the actual contact point as determined by the above procedure is in opposition to the tangential deflection, sliding stops. Since the tangential deflection has been maintained consistent with the tangential force during sliding, a smooth transition takes place when sliding stops.

This completes the development of the tire forces. The total tire force, comprised of F_N^δ, $F_N^{\dot{\delta}}$, F_T^δ, and $F_T^{\dot{\delta}}$, all acting in their appropriate directions, will be designated by F_t.

SECTION 3

AERODYNAMIC LOADING

3.1 Introduction

The loading on the vehicle results from the blast wave from a nuclear explosion. Associated with the blast wave are increased pressure and density and material velocity. Blast wave characteristics for two heights of burst are contained in the program. Those corresponding to a height of burst of zero are based upon curve fits in Reference 1; those for a scaled burst height of 60 $W^{1/3}$ meters are based upon Reference 3.

The blast wave front is assumed to be normal to the ground surface, and the vehicle is assumed initially to be essentially normal to the ground. As the blast wave envelops the vehicle, the wave reflects from the vehicle surface and rarefaction waves emanate from free edges to relieve the reflected pressure. At later times, the loading is essentially a drag type loading, resulting from the material velocity associated with the blast wave. The development of the aerodynamic loading is separated into these two regimes, which will be referred to as the diffraction loading and the drag loading.

3.2 Diffraction Loading

The diffraction loading is based upon shock-tube experiments reported in Reference 4. In these experiments, front and back face pressures on a rectangular block were measured with the shock wave normally incident on the front face (shock front parallel to front face). For the present application, diffraction loadings for arbitrary intercept angles are required. To permit extension of the shock tube results to arbitrary intercept angles, a crude model roughly reproducing the diffraction processes has been constructed. The model is fitted to the experimental front and back face pressures and, since the physical processes are at least roughly modeled, it is hoped that reasonable pressures are predicted for other intercept angles.

It is assumed that the vehicle moves very little during the diffraction period, so that the shock wave and the vehicle may both be taken as normal to the ground. This assumption limits the number of configurations which must be addressed. The intercept geometry is shown in Figure 3. It may be seen that $\theta = 0^{\circ}$ corresponds to the shock-tube experiment, $\theta = 90^{\circ}$ corresponds to intercept of the vehicle from the front, and $\theta = -90^{\circ}$ corresponds to intercept from the rear. Due to truck symmetry, only the range $-90^{\circ} \leq \theta \leq 90^{\circ}$ need be considered.

---

[3] Ethridge, Noel H., private communication.

[4] Taylor, W. J., A Method for Predicting Blast Loads During the Diffraction Phase, Shock and Vibration Bulletin 42, Part 4, pg. 135, January, 1972.

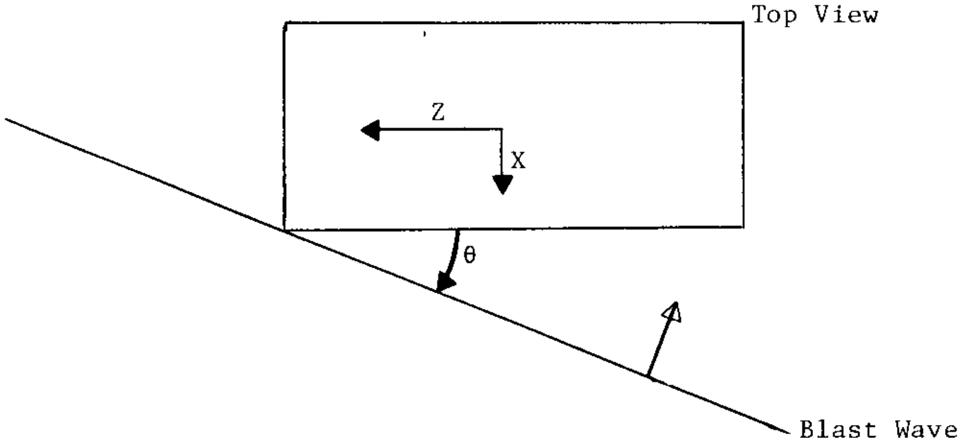

FIGURE 3. BLAST WAVE INTERCEPT GEOMETRY

The basic model for the diffraction loading involves waves emanating from free edges. For a point, p, on a rectangular face, four distances are defined, as shown in Figure 4. A wave is assumed to emanate from each free edge at the time when the undisturbed shock front reaches the edge point defined by s_i. The wave moves at a velocity "a", which will be defined later. With the shock velocity designated as V_s, effective edge distances, $\bar{s}_i$, may be defined which account for the time at which the wave starts relative to the time at which the undisturbed shock reaches point p. For example, suppose that the face shown in Fig. 4 is the left side of the vehicle, which is the side first intercepted by the shock wave. Suppose that θ is positive, so that the edge defined by s_1 is intercepted first, and define zero time as the time of first intercept.

The time at which the undisturbed shock reaches point p is $\frac{s_1 \sin\theta}{V_s}$.

At that time, the rarefaction wave from the s_1 free edge will have moved a distance $s_1 \frac{a \sin\theta}{V_s}$ toward point p. Hence, in effect the edge distance s_1 is decreased by the amount $s_1 \frac{a \sin\theta}{V_s}$. The effective edge distance in this case is then

$$\bar{s}_1 = s_1 \; [1 - \frac{a \sin\theta}{V_s}] \tag{24a}$$

Similarly, for the same left side face,

$$\bar{s}_2 = s_2 \tag{24b}$$

$$\bar{s}_3 = s_3 \; [1 + \frac{a \sin\theta}{V_s}] \tag{24c}$$

$$\bar{s}_4 = s_4 \tag{24d}$$

The other faces may be dealt with similarly. For the front of the vehicle, with Fig. 4 considered as a view from in front,

$$\bar{s}_1 = s_1 \; [1 + \frac{a \cos\theta}{V_s}] \tag{25a}$$

$$\bar{s}_2 = s_2 \tag{25b}$$

$$\bar{s}_3 = s_3 \; [1 - \frac{a \cos\theta}{V_s} \tag{25c}$$

$$\bar{s}_4 = s_4 \tag{25d}$$

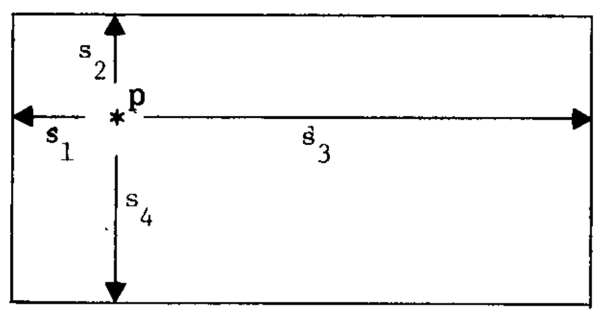

FIGURE 4. DIFFRACTION LOADING MODEL

For the rear of the vehicle, with Fig. 4 considered as a view from the rear,

$$\bar{s}_1 = s_1 \quad [1 - \frac{a \cos\theta}{V_s}] \tag{26a}$$

$$\bar{s}_2 = s_2 \tag{26b}$$

$$\bar{s}_3 = s_3 \quad [1 + \frac{a \cos\theta}{V_s}] \tag{26c}$$

$$\bar{s}_4 = s_4 \tag{26d}$$

For the right side of the vehicle, with Fig. 4 considered as a view from the left side, Eqs. 24 apply. The top and the bottom of the vehicle are treated separately later.

The wave from the free edge is assumed to produce an exponentially varying pressure at point p with time. Hence, the form of the diffraction loading is taken as

$$\Delta p(t) = [\Delta p(t=0) - \Delta p(t=\infty)] e^{-k \sum_{i=1}^{4} E_i (\tau_i - 1)} + \Delta p(t=\infty) \tag{27}$$

The time zero reference is the time at which the undisturbed shock reaches point p. The quantities $\Delta p(t=0)$ and $\Delta p(t=\infty)$ are the pressures at point p at time zero plus and time equal to infinity, respectively. These will be defined later. The parameter E_i is an effectiveness factor, which will be defined later also. For the moment, consider E_i to be unity. The dimensionless time parameter, τ_i, is defined as

$$\tau_i \equiv \frac{a\,t}{\bar{s}_i} \tag{28}$$

A wave from the edge is seen to reach point p at $\tau_i = 1$. For $\tau_i < 1$, $(\tau_i - 1)$ in Eq. 27 is taken as zero.

The constant k is seen to be the only free variable available to fit the shock tube data. The empirical fit for k is as follows.

Defining the shock strength

$$\xi \equiv \frac{\Delta p_s}{p_o} \tag{29}$$

where Δp_s is the shock overpressure and p_o is the ambient pressure, and defining

$$\bar{k} = 0.1 + \frac{0.07}{\xi} \tag{30}$$

the constant k is given by

$$\begin{aligned} k &= \bar{k} \quad \text{for} \quad 0 \le \alpha \le 90^{\circ} \\ k &= \bar{k} \quad (1-0.2\xi) \text{ for } \alpha = 180^{\circ} \end{aligned} \tag{31}$$

The angle α is the angle between the shock wave and the vehicle surface being considered. For example, for $\theta = 0^{\circ}$, α for the left side is 0°, for the right side 180°, and for the front and back, 90°. For $\theta = 45^{\circ}$, α is 45° for the left side, 135° for the right side, 45° for the front, and 135° for the back. For $90 < \alpha < 180^{\circ}$, linear interpolation between k ($\alpha = 90^{\circ}$) and k ($\alpha = 180^{\circ}$) is used. The values of α corresponding to the shock tube experiment are 0° and 180°.

The pressure at time zero plus is the reflected pressure for surfaces facing the blast wave and zero for surfaces facing away from the blast wave. The reflected pressure versus angle of incidence is taken from Reference 5, and is defined in Fig. 5 for $\alpha \le 90^{\circ}$. For $\alpha > 90^{\circ}$, $\Delta p(t=0)$ is taken as zero.

The pressure at time equal to infinity should be simply the drag phase pressure, $\Delta p_s + c_p\, q$, where c_p is the pressure coefficient and q is the dynamic pressure. However, in order to ensure transition from the diffraction period loading to the drag phase loading, $\Delta p(t=\infty)$ is biased from this value. For positive pressure coefficients,

$$\Delta p(t=\infty) = \Delta p_s \quad \text{for} \quad c_p > 0 \tag{32}$$

[5] Lee, William N. and Mente, Lawrence J., NOVA-2--A Digital Computer Program for Analyzing Nuclear Overpressure Effects on Aircraft, Part 1, Theory, AFWL-TR-75-262, Pt 1, August, 1976.

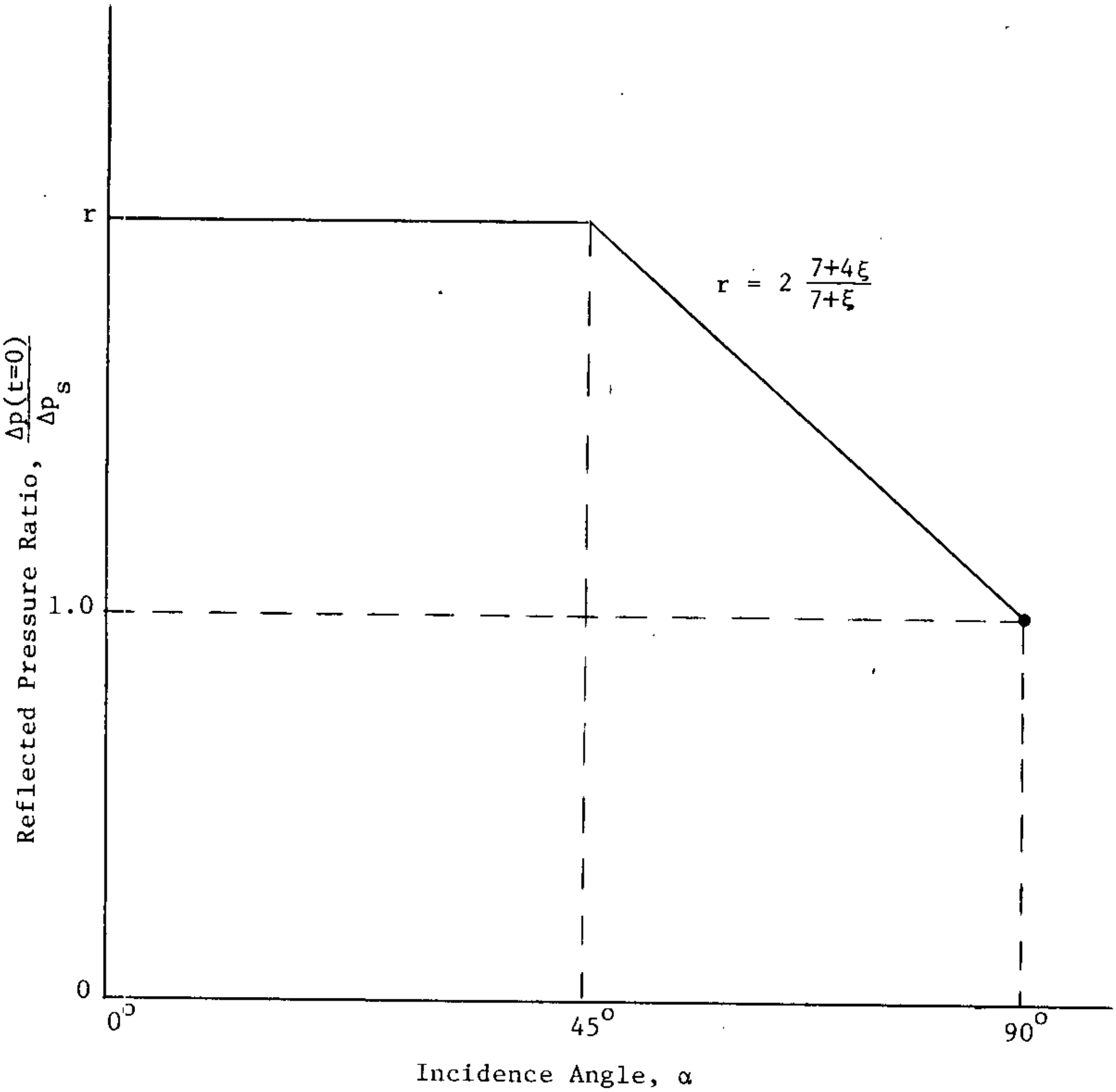

FIGURE 5. REFLECTED PRESSURE

Transition to drag phase loading takes place when the diffraction loading becomes less than the drag loading.

For negative c_p and $\alpha \leq 90^o$,

$$\Delta p(t=\infty) = \Delta p_s + 2\ c_p\ q \text{ for } c_p < 0 \quad (33)$$

Transition to drag phase loading again takes place when the diffraction loading becomes less than the drag loading.

Finally, for $\alpha > 90^o$ (c_p is always negative for this case),

$$\Delta p\ (t=\infty) = \Delta p_s \quad \alpha > 90^o \quad (34)$$

Transition to drag phase loading takes place when the diffraction loading becomes greater than the drag loading.

Returning to the definition of the speed of the waves from the free edges, a, the first step is to define the flow velocity immediately behind the incident shock wave pattern. For a weak wave, the flow velocity along the surface, W, is given by

$$W = 2\ w \sin \alpha \quad (35)$$

where w is the material velocity behind the shock wave. For $\alpha = 90^o$, W must be equal to w, which is inconsistent with Eq. 35. In order to satisfy the known point at $\alpha = 90^o$, Eq. 35 is rather arbitrarily replaced by Fig. 6, which is then used to define the flow velocity. For $\alpha > 90^o$, w is taken as zero.

The material velocity, w, and the shock velocity, V_s, are given in terms of the shock strength, ξ, through the Rankine-Hugoniot relations.

$$w = \frac{5\ \xi}{\sqrt{7(7+6\ \xi}}\ a_o \quad (36)$$

$$V_s = \sqrt{\frac{7+6\ \xi}{7}}\ a_o \quad (37)$$

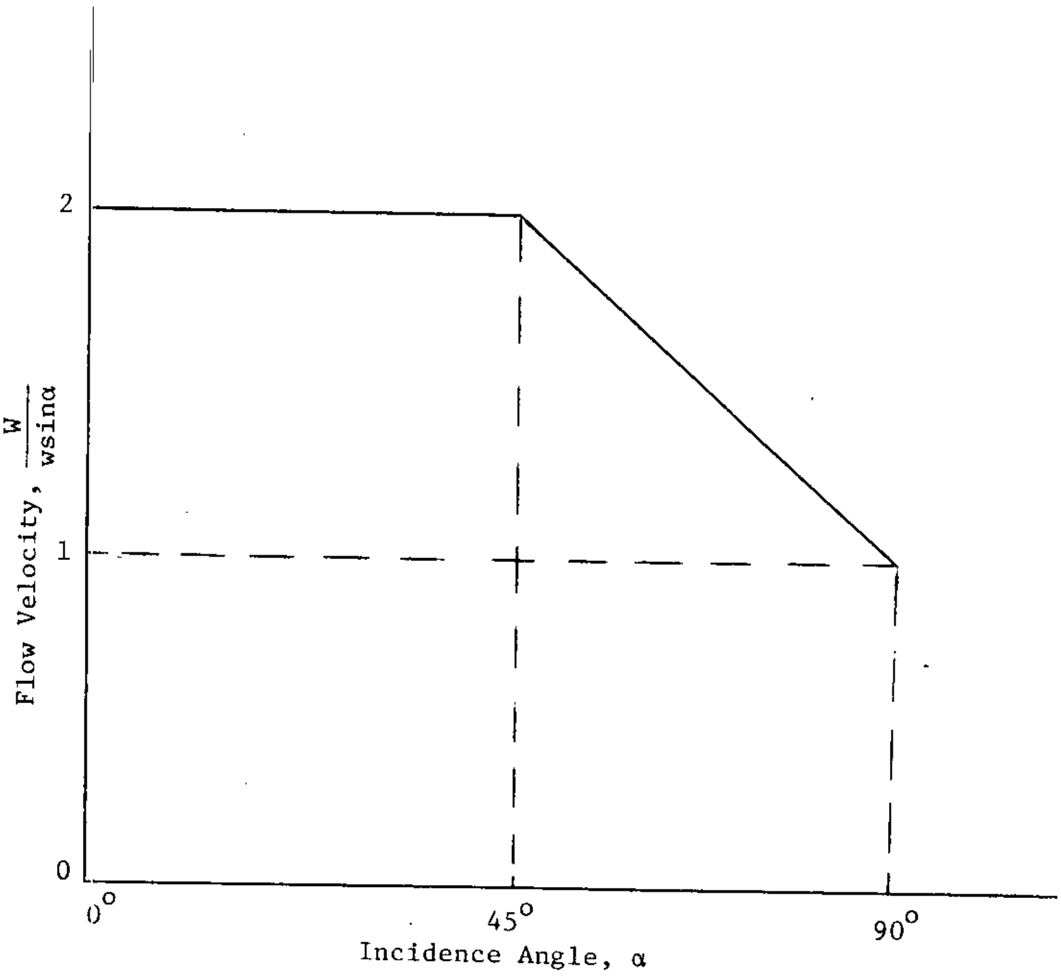

FIGURE 6. FLOW VELOCITY

where a_o is the ambient speed of sound. The speed of sound behind the incident shock wave pattern, a_5, is

$$a_5 = \sqrt{\frac{(8\xi+7)(2\xi+7)}{7(7+6\xi)}}\ a_o \quad \text{for} \quad \alpha=0^o$$

$$a_5 = \sqrt{\frac{(\xi+1)(\xi+7)}{7+6\xi}}\ a_o \quad \text{for} \ \alpha=90^o \tag{38}$$

For other angles, a_5 is approximated as in Figure 7.

The preceding relations allow the definition of the wave speed "a" for the various incidence angles and surfaces. These definitions are given in Table 2. Note that θ has been interpreted in terms of the corresponding value of α.

The development up to this point completely defines the diffraction phase loading for the shock tube experiments, since E_i in Eq. 27 is unity for those cases. The model described herein agrees with the experimental data within experimental accuracy. It should be recalled, however, that the shock tube experiment was solely for $\theta = 0^o$, and was also for a single block geometry. Accordingly it must be remembered that the present model is of unknown accuracy for other geometries and incidence angles, particularly for other incidence angles.

The parameter E_i requires general definition. The need for such a parameter, which is essentially an edge effectiveness factor, may be seen from the following considerations. Suppose $\theta = 45^o$. The shock then strikes the front and left side faces equally. Hence, no clearing wave emanates from the corner first struck by the shock wave in either direction. But the definition of the relevant $\bar{s}_i$ leads to small or even negative values of $\bar{s}_i$ (see Eq. 24a). Since the edge is actually completely ineffective, E_i should be assigned a value of zero for this case. Also, E_i should be unity for $\alpha=0^o$, to maintain the correlation with the shock tube results. For $\alpha>90^o$, there is no problem with the definition of the edge distances and E_i should thus be unity. For α between 45^o and 90^o, the pressure on the relevant face is higher than that on the adjacent face; hence, the edge is ineffective and E_i must be taken as zero. Finally, unless E_i is taken as zero for α slightly less than 45^o, s_i will go to zero at moderate shock pressure values, and will thus give unreasonable results. Rather arbitrarily, E_i has been taken as going to zero at $\alpha=30^o$, varying linearly from unity at $\alpha=0^o$ to that value at $\alpha=30^o$. E_i is zero from $\alpha=30^o$ to $\alpha=90^o$, and is unity for $\alpha>90^o$.

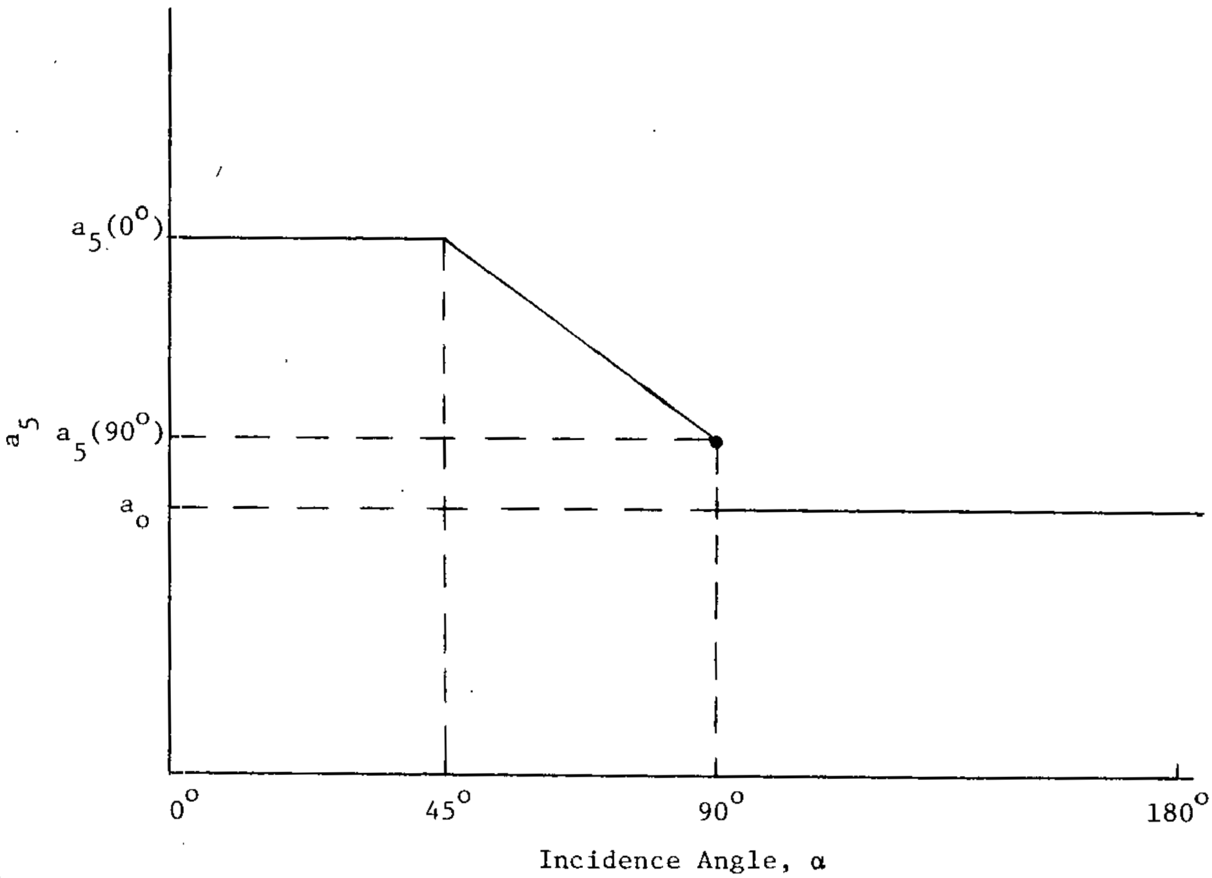

FIGURE 7. SPEED OF SOUND BEHIND SHOCK PATTERN

TABLE 2

Definition of Wave Speed, a

Surface of Vehicle	θ	i=1	i=2	i=3	i=4
Left Side	$\geq 0^{\circ}$	$a_5(\theta)+W(\theta)$	$a_5(\theta)$	$a_5(\theta)-W(\theta)$	$a_5(\theta)$
	$< 0^{\circ}$	$a_5(\lvert\theta\rvert)-W(\lvert\theta\rvert)$	$a_5(\lvert\theta\rvert)$	$a_5(\lvert\theta\rvert)+W(\lvert\theta\rvert)$	$a_5(\lvert\theta\rvert)$
Right Side	$+90^{\circ}$	$a_5(90^{\circ})+w$	$a_5(90^{\circ})$	$a_5(90^{\circ})-w$	$a_5(90^{\circ})$
	$-90^{\circ}<\theta<90^{\circ}$	a_o	a_o	a_o	a_o
	-90°	$a_5(90^{\circ})-w$	$a_5(90^{\circ})$	$a_5(90^{\circ})+w$	$a_5(90^{\circ})$
Front	$0^{\circ}\leq\theta\leq 90^{\circ}$	$a_5(90-\theta)-W(90-\theta)$	$a_5(90-\theta)$	$a_5(90-\theta)+W(90-\theta)$	$a_5(90-\theta)$
	$-90^{\circ}\leq\theta<0^{\circ}$	a_o	a_o	a_o	a_o
Rear	$0<\theta\leq 90^{\circ}$	a_o	a_o	a_o	a_o
	$-90^{\circ}\leq\theta\leq 0^{\circ}$	$a_5(90+\theta)+W(90+\theta)$	$a_5(90+\theta)$	$a_5(90+\theta)-W(90+\theta)$	$a_5(90+\theta)$

The next problem requiring attention is the definition of s_i for non-rectangular shapes. The development up to this point has been completely concerned with rectangular faces. The extension to non-rectangular faces is not obvious. The convention adopted herein is depicted in Fig. 8. The surface treated is assumed to consist of a group of rectangular surfaces. For the cross-hatched areas, the definition of the s_i is consistent with that given earlier. For a point p in the non-cross-hatched area, s_1' and s_2' are defined in the manner in which s_1 and s_2 would ordinarily be defined. In addition, the distance r to the interior corner is defined as indicated in Fig. 8. The larger of s_1' and s_2' is accepted as s_1 or s_2; in Fig. 8, s_1' will be taken as s_1. The smaller of the remaining s_i' or r is taken as the other s_i; in Fig. 8, r will be taken as s_2.

Finally the top and bottom surfaces have not been treated yet. The treatment adopted herein is simply to use the drag phase loading always. This is in error at time zero, when the pressure should be Δp_s and is instead $\Delta p_s + c_p q$. The difference is small, however, because c_p is small. The real problem with the top and bottom surfaces is that the most significant effect on overall motion is produced by the difference between the top and bottom surface loadings. The bottom surface loading cannot be predicted well, due to the unevenness of the bottom surface, ground interference, and so forth. Hence, an approximation to the top and bottom surface loadings is acceptable, and the exclusive use of the drag loading is simple and expedient.

The foregoing has defined the pressure at a point. Integration over all points then defines the loading on a surface. In practice, of course, the integration is replaced by a summation over several points, with each point used as representative of some portion of the surface area.

3.3 Drag Loading

Following the diffraction phase, the pressure loading becomes a drag type of loading. As indicated in the previous division, the pressure loading during the drag phase is given by

$$\Delta p(t) = \Delta p_s + c_p q \tag{39}$$

where Δp_s and q are the overpressure and dynamic pressure associated with the blast wave. The task of estimating the drag-phase loading thus becomes a matter of determining the pressure coefficient, c_p. The pressure coefficient must be determined for arbitrary burst orientations (with the shock assumed to be traveling parallel to the ground) for the various rectangular areas which make up the exterior surfaces of the vehicle model.

These are numerous factors which can affect the value of c_p; however, most of these will have to be ignored, as will be seen. In establishing reasonable values of c_p for this application, use is made

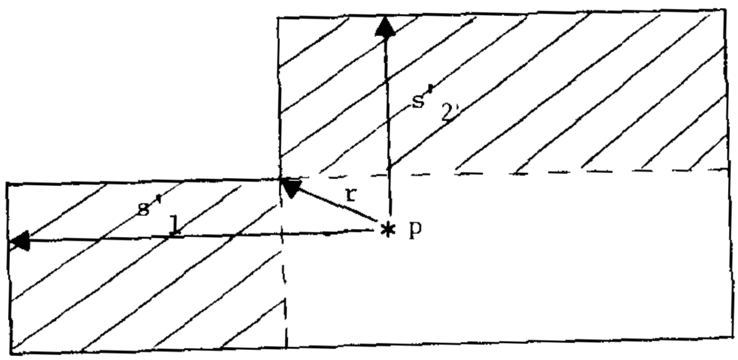

FIGURE 8. DEFINITION OF s_i FOR NON-RECTANGULAR FACE

of available experimental data for c_p and for the "drag coefficient" C_D on box-like bodies. C_D represents a difference in pressure between opposing surfaces of a structure.

Some of the parameters which affect the value of c_p are Reynolds number, corner radii, Mach number, aspect ratio of body being studied, ground effects, and the angle between the flow velocity and the surface normal, α. Of these, only the variation with α will be considered. It is rather obvious that α is the most important of the above parameters with respect to affecting c_p, and must be accounted for. Of the remaining parameters, the most important are the Reynolds number and the corner radii, which affect the critical Reynolds number. The Reynolds number varies with time as the blast wave decays, and is also, of course, a function of blast wave strength. Since the largest drag loading occurs early while the blast wave characteristics are near their maxima, the selection of the pressure coefficient has been based on Reynolds numbers in the vicinity of maximum Reynolds numbers associated with blast wave strengths between about 2 and 10 psi at sea level. Mach number has little effect for the rather small Mach numbers associated with the above blast strengths. Aspect ratio has a modest effect and cannot easily be accounted for for the odd shapes to be addressed. Ground effects are much too complex to be handled in the present development, and are undoubtedly less important than Reynolds number and corner radii effects.

Thus, it remains to determine c_p as a function of α. The angle α is the angle between the material velocity vector and the inward facing surface normal. This is the same as the angle α defined in the previous division, except that there, in considering the diffraction loading, the vehicle was assumed not to move. In the drag loading definition, the instantaneous orientation of the vehicle is used, so that the present α may be considered to be a generalization of the α used in defining the diffraction phase loading.

First consider a block with the shock front parallel to one face ($\alpha=0^o$). For the opposite face, $\alpha = 180^o$. References 6 and 7 indicate that an average value of 1.25 for C_D is reasonable. At the same time, a good average value of c_p for the leeward surface seems to be about -0.4 Hence, since C_D is related to a pressure differential, a value of 0.85 for c_p is adopted for the windward surface.

Having assumed -0.4 and 0.85 as values of c_p on the leeward and windward sides ($\alpha=0^o$ and $\alpha=180^o$), it is necessary to extend the formulation for arbitrary values of α. Referring to experimental evidence in Reference 7 again, there is indication that the value of c_p decreases by approximately a function of $\cos\alpha$ for $\alpha \leq 30^o$ but thereafter somewhat faster until a value of approximately -0.4 or -0.5 is reached at about 105^o-110^o. Thus, the following approximate formula is adopted:

---

[6]Hoerner, S. F., "Fluid-Dynamic Drag", 1965.

[7]Hankins, Dorris M., Experimental Pressure Distributions and Force Coefficients on Block Forms for Varying Mach Number, Reynolds Number, and Yaw Angle, SC-4204(TR), January, 1959.

$$c_p = \begin{cases} 0.85 \cos\left(\frac{9}{8}\alpha\right) & \left(0 \leq \alpha \leq \frac{7\pi}{12}\right. \\ -0.4 & \left(\frac{7\pi}{12} < \alpha \leq \pi\right) \end{cases} \tag{40}$$

SECTION 4
EQUATIONS OF MOTION

4.1 Introduction

In the preceding sections, the mass model, the generalized coordinates, and the forces from the springs, dampers, and tires have been introduced. In order to obtain the equations of motion, the inertia terms must be defined and the various forces must be converted to generalized forces.

4.2 Inertia Terms

The body-axis components of the acceleration of a mass are

$$\begin{aligned} a_x = {} & \ddot{g}_2 - \dot{g}_1\dot{g}_3 + \dot{g}_5\dot{g}_6 + \ddot{L}_x - L_y\ddot{g}_1 + L_z\ddot{g}_5 \\ & -2\dot{L}_y\dot{g}_1 + 2\dot{L}_z\dot{g}_5 - L_x(\dot{g}_1^2 + \dot{g}_5^2) + \dot{g}_4(L_y\dot{g}_5 + L_z\dot{g}_1) \end{aligned} \qquad (41a)$$

$$\begin{aligned} a_y = {} & \ddot{g}_3 - \dot{g}_4\dot{g}_6 + \dot{g}_1\dot{g}_2 + \ddot{L}_y + L_x\ddot{g}_1 - L_z\ddot{g}_4 + 2\dot{L}_x\dot{g}_1 - 2\dot{L}_z\dot{g}_4 \\ & - L_y(\dot{g}_1^2 + \dot{g}_4^2) + \dot{g}_5(L_x\dot{g}_4 + L_z\dot{g}_1) \end{aligned} \qquad (41b)$$

$$\begin{aligned} a_z = {} & \ddot{g}_6 + \dot{g}_4\dot{g}_3 - \dot{g}_5\dot{g}_2 + \ddot{L}_z - L_x\ddot{g}_5 + L_y\ddot{g}_4 - 2\dot{L}_x\dot{g}_5 + 2\dot{L}_y\dot{g}_4 \\ & - L_z(\dot{g}_4^2 + \dot{g}_5^2) + \dot{g}_1(L_x\dot{g}_4 + L_y\dot{g}_5) \end{aligned} \qquad (41c)$$

These equations may be found in Reference 8, although with different variable names. As before, a dot indicates differentiation with respect to time. The time derivatives of the lengths may easily be defined in terms of the time derivatives of the generalized coordinates in Table 1.

---

[8] Hobbs, Norman P., Zartarian, Garabed, and Walsh, John P., <u>A Digital Computer Program for Calculating the Blast Response of Aircraft to Nuclear Explosions, Volume 1, Program Description</u>, AFWL-TR-70-140, Vol. 1, April, 1971.

The inertia terms in the equations of motion may be defined by taking the dot product of the acceleration vector and the vector which defines the displacement of a mass due to a unit displacement of a generalized coordinate, and by then integrating the mass times this dot product over all the masses. For the generalized coordinate g_1, for example, the displacement of a mass due to a unit displacement has the components $(-L_y, L_x, 0)$. The corresponding inertia term is thus

$$\int \left\{ -L_y[\ddot{g}_2 - \dot{g}_1\dot{g}_3 + \dot{g}_5\dot{g}_6 + \ddot{L}_x - L_y\ddot{g}_1 + L_z\ddot{g}_5 - 2\dot{L}_y\dot{g}_1 + 2\dot{L}_z\dot{g}_5 - L_x(\dot{g}_1^2 + \dot{g}_5^2) + \dot{g}_4\,(L_y\dot{g}_5 + L_z\dot{g}_1)] + L_x[\ddot{g}_3 - \dot{g}_4\dot{g}_6 + \dot{g}_1\dot{g}_2 + \ddot{L}_y + L_x\ddot{g}_1 - L_z\ddot{g}_4 + 2\dot{L}_x\dot{g}_1 - 2\dot{L}_z\dot{g}_4 - L_y(\dot{g}_1^2 + \dot{g}_4^2) + \dot{g}_5\,(L_x\dot{g}_4 + L_z\dot{g}_1)] \right\} dm$$

Recalling that the lengths, L, are measured from the vehicle center of gravity and that symmetry exists about the y-z plane, and introducing the discrete masses, M, and their inertias, I, about their own centers of gravity, considerable simplification is possible, leading to

$$\sum \left\{ [I_{zz} + M(L_x^2 + L_y^2)]\,\ddot{g}_1 - [I_{yz} + ML_yL_z]\ddot{g}_5 - ML_y\,\ddot{g}_x + ML_x\ddot{g}_y + I_{zz}\ddot{g}_\psi + [I_{yy} - I_{xx} + M(L_x^2 - L_y^2)]\dot{g}_4\dot{g}_5 - [I_{yz} + ML_yL_z]\dot{g}_1\dot{g}_4 - 2M\dot{g}_z[L_x\dot{g}_4 + L_y\dot{g}_5] + 2M\dot{g}_1\,[L_y\dot{g}_y + L_x\dot{g}_x] - \dot{g}_5[I_{xx} + I_{zz} - I_{yy}]\dot{g}_\phi + \dot{g}_4[I_{yy} + I_{zz} - I_{xx}]\dot{g}_\theta \right\}$$

The assumption that the product of inertia, I_{yz}, is non-zero only for the truck body mass has been used in the above expression. In addition, substitutions such as $\dot{L}_x = \dot{g}_x - \ell_y \dot{g}_\psi + \ell_z \dot{g}_\theta$ have been made, where it will be recalled that the length ℓ is measured from the center of gravity of the individual mass.

From the above expression, it is apparent that the inertia terms may be separated into two parts; those involving second derivatives and those not involving second derivatives. The coefficients of the second derivative terms form the generalized mass matrix, which is given in Table 3. Because of the use of body axes and the assumption of small distortions, the mass matrix is constant. The terms not involving second derivatives are called inertia forces here, and these are listed in Table 4.

4.3 Generalized Forces

The forces from the springs and dampers and tires have been derived in Section 2 and the aerodynamic loading was developed in Section 3. These forces need to be converted to generalized forces for use in the equations of motion.

The spring/damper forces will be designated as F_s. These forces are in the directions of the springs or dampers. In Division 2.3, a λ transformation matrix was defined which converts generalized coordinate displacements into spring elongations. The generalized force may be found by taking the dot product of the spring force vector with the spring displacement vector corresponding to a unit displacement of the generalized coordinate. Thus, the generalized force, GF, becomes simply

$$[GF_s] = [F_s]\ [\lambda] \tag{42}$$

It will be recalled that λ contains no elements corresponding to the first six degrees of freedom; that is, corresponding to the generalized coordinates for overall vehicle motion. Hence, the spring generalized force for each of the first six coordinates is zero, which is physically obvious since an internal force cannot produce overall vehicle motion.

The tire forces derived in Division 2.4 are in body-axis components, and are designated by F_t. The conversion to generalized forces is obvious from the above considerations, and the results are given in Table 5.

The aerodynamic forces from Section 3, F_a, are also in body-axis components. The associated generalized forces are given in Table 6.

The generalized forces due to gravity may be expressed in terms of the β transformation matrix which relates earth-fixed axis motions to body-axis motions.

TABLE 3

GENERALIZED MASS MATRIX

$$M_{1,1} = \sum [I_{zz} + M\,(L_x^2 + L_y^2)]$$

$$M_{1,5} = -\sum [I_{yz} + ML_y L_z]$$

$$M_{1,g_\psi} = I_{zz}$$

$$M_{1,g_x} = -ML_y$$

$$M_{1,g_y} = ML_x$$

$$M_{2,2} = \sum M$$

$$M_{2,g_x} = M$$

$$M_{3,3} = \sum M$$

$$M_{3,g_y} = M$$

$$M_{4,4} = \sum [I_{xx} + M\,(L_y^2 + L_z^2)]$$

$$M_{4,g_y} = -ML_z$$

$$M_{4,g_\phi} = I_{xx}$$

$$M_{4,g_z} = ML_y$$

TABLE 3 (CONT'D.)

GENERALIZED MASS MATRIX

$$M_{5,5} = \sum [I_{yy} + M\ (L_x^2 + L_z^2)]$$

$$M_{5,g_x} = ML_z$$

$$M_{5,g_\theta} = I_{yy}$$

$$M_{5,g_z} = -ML_x$$

$$M_{6,6} = \sum M$$

$$M_{6,g_z} = M$$

$$M_{g_\psi,g_\psi} = I_{zz}$$

$$M_{g_x,g_x} = M$$

$$M_{g_y,g_y} = M$$

$$M_{g_\phi,g_\phi} = I_{xx}$$

$$M_{g_\theta,g_\theta} = I_{yy}$$

$$M_{g_z,g_z} = M$$

Note that $M_{j,i} = M_{i,j}$ and that $M_{i,j} = 0$ if not listed.

Summations extend over all masses.

Where there is no summation, the mass or inertia is that associated with the particular degree of freedom.

TABLE 4

INERTIA FORCES

$$F_{I_1} = \sum \Big\{ [I_{yy} - I_{xx} + M(L_x^2 - L_y^2)]\dot{g}_4\dot{g}_5$$

$$-[I_{yz} + M\,L_yL_z]\dot{g}_1\dot{g}_4 - 2M\dot{g}_z[L_x\dot{g}_4 + L_y\dot{g}_5]$$

$$+2M\dot{g}_1[L_y\dot{g}_y + L_x\dot{g}_x] - [I_{xx}+I_{zz}-I_{yy}]\dot{g}_5\dot{g}_\phi$$

$$+ \ [I_{yy}+I_{zz}-I_{xx}]\dot{g}_4\dot{g}_\theta \Big\}$$

$$F_{I_2} = \sum M\Big\{ -\dot{g}_1[\dot{g}_3 + 2\dot{g}_y] + \dot{g}_5\ [\dot{g}_6 + 2\dot{g}_z] \Big\}$$

$$F_{I_3} = \sum M\Big\{ \ \dot{g}_1[\dot{g}_2 + 2\dot{g}_x] - \dot{g}_4\ [\dot{g}_6 + 2\dot{g}_z] \Big\}$$

$$F_{I_4} = \sum \Big\{ \ [I_{zz}-I_{yy} + M(L_y^2-L_z^2)]\dot{g}_1\dot{g}_5$$

$$+[I_{yz}+ML_yL_z](\dot{g}_1^2-\dot{g}_5^2)-2M\dot{g}_x[L_z\dot{g}_1 + L_y\dot{g}_5]$$

$$+ 2M\dot{g}_4[L_y\dot{g}_y + L_z\dot{g}_z] - [I_{xx}+I_{yy}-I_{zz}]\dot{g}_1\dot{g}_\theta$$

$$+ [I_{xx}+I_{zz} - I_{yy}]\dot{g}_5\dot{g}_\psi \Big\}$$

$$F_{I_5} = \sum \Big\{ \ [I_{xx}-I_{zz} + M(L_z^2 - L_x^2)]\dot{g}_1\dot{g}_4$$

$$+[I_{yz}+ML_yL_z]\dot{g}_4\dot{g}_5-2M\dot{g}_y[L_z\dot{g}_1+L_x\dot{g}_4]$$

$$+ 2M\dot{g}_5[L_x\dot{g}_x + L_z\dot{g}_y]+ [I_{xx}+I_{yy}-I_{zz}]\dot{g}_1\dot{g}_\phi$$

$$- [I_{yy}+I_{zz} - I_{xx}]\dot{g}_4\dot{g}_\psi \Big\}$$

TABLE 4 (CONT'D.)

INERTIA FORCES

$$F_{I_6} = \sum M \left\{ \dot{g}_4[\dot{g}_3 + 2\dot{g}_y] - \dot{g}_5 \; [\dot{g}_2 + 2\dot{g}_x] \right\}$$

$$F_{I_\psi} = [I_{yy} - I_{xx}]\dot{g}_4\dot{g}_5 - [I_{xx} + I_{zz} - I_{yy}]\dot{g}_5\dot{g}_\phi$$

$$+ [I_{yy} + I_{zz} - I_{xx}]\dot{g}_4\dot{g}_\theta$$

$$F_{I_x} = M \left\{ - \dot{g}_1[\dot{g}_3 + 2\dot{g}_y] + \dot{g}_5 \; [\dot{g}_6 + 2\dot{g}_z] - L_x[\dot{g}_1^2 + \dot{g}_5^2] \right.$$

$$\left. + \dot{g}_4[L_y\dot{g}_5 + L_z\dot{g}_1] \right\}$$

$$F_{I_y} = M \left\{ \dot{g}_1[\dot{g}_2 + 2\dot{g}_x] - \dot{g}_4 \; [\dot{g}_6 + 2\dot{g}_z] - L_y \; [\dot{g}_1^2 + \dot{g}_4^2] \right.$$

$$\left. + \dot{g}_5[L_x\dot{g}_4 + L_z\dot{g}_1] \right\}$$

$$F_{I_\phi} = [I_{zz} - I_{yy}]\dot{g}_1\dot{g}_5 - [I_{xx} + I_{yy} - I_{zz}]\dot{g}_1\dot{g}_\theta$$

$$+ [I_{xx} + I_{zz} - I_{yy}]\dot{g}_5\dot{g}_\psi$$

$$F_{I_\theta} = [I_{xx} - I_{zz}]\dot{g}_1\dot{g}_4 + [I_{xx} + I_{yy} - I_{zz}]\dot{g}_1\dot{g}_\phi$$

$$- [I_{yy} + I_{zz} - I_{xx}]\dot{g}_4\dot{g}_\psi$$

$$F_{I_z} = M \left\{ \dot{g}_4[\dot{g}_3 + 2\dot{g}_y] - \dot{g}_5 \; [\dot{g}_2 + 2\dot{g}_x] - L_z[\dot{g}_4^2 + \dot{g}_5^2] \right.$$

$$\left. + \dot{g}_1[L_x\dot{g}_4 + L_y\dot{g}_5] \right\}$$

TABLE 5

GENERALIZED FORCES FROM TIRE FORCES

$$GF_{t_1} = -(L_y - R)F_{t_x} + L_x F_{t_y}$$

$$GF_{t_2} = F_{t_x}$$

$$GF_{t_3} = F_{t_y}$$

$$GF_{t_4} = -L_z F_{t_y} + (L_y - R)F_{t_z}$$

$$GF_{t_5} = L_z F_{t_x} - L_x F_{t_z}$$

$$GF_{t_6} = F_{t_z}$$

$$GF_{t_\psi} = -(\ell_y - R)F_{t_x} + \ell_x F_{t_y} \,{}^*$$

$$GF_{t_y} = F_{t_y}$$

Note: Since the axles have only ψ and y degrees of freedom, GF_{t_x}, GF_{t_ϕ}, GF_{t_θ}, and GF_{t_z} are not required.

* For a tracked vehicle with no axle connecting the two tires, the F_{t_x} contribution must be discarded; hence, $GF_{t_x} = \ell_x F_{t_y}$ for a tracked vehicle.

TABLE 6

GENERALIZED FORCES FROM AERODYNAMIC LOADS

$$GF_{a_1} = \sum_{\text{all}} [-L_y F_{a_x} + L_x F_{a_y}]$$

$$GF_{a_2} = \sum_{\text{all}} F_{a_x}$$

$$GF_{a_3} = \sum_{\text{all}} F_{a_y}$$

$$GF_{a_4} = \sum_{\text{all}} [-L_z F_{a_y} + L_y F_{a_z}]$$

$$GF_{a_5} = \sum_{\text{all}} [L_z F_{a_x} - L_x F_{a_z}]$$

$$GF_{a_6} = \sum_{\text{all}} F_{a_z}$$

$$GF_{a_\psi} = \sum_{\text{shelter}} [-\ell_y F_{a_x} + \ell_x F_{a_y}]$$

$$GF_{a_y} = \sum_{\text{shelter}} F_{a_y}$$

$$GF_{a_\phi} = \sum_{\text{shelter}} [-\ell_z F_{a_y} + \ell_y F_{a_z}]$$

$$GF_{a_z} = \sum_{\text{shelter}} F_{a_z}$$

Note: Aerodynamic loads do not act on the racks, which are presumed to be inside the shelter. Aerodynamic loads on the tires are allowed only in the x direction, and the axles do not have an x degree of freedom. Hence, generalized forces from the aerodynamic loads exist only for the overall vehicle degrees of freedom and the shelter degrees of freedom.

$$\begin{Bmatrix} GF_{g_x} \\ GF_{g_y} \\ GF_{g_z} \end{Bmatrix} = - Mg \begin{Bmatrix} \beta_{2,1} \\ \beta_{2,2} \\ \beta_{2,3} \end{Bmatrix} \tag{43}$$

For overall vehicle degrees of freedom, summation over all masses is required. For individual masses, of course, only the relevant mass is used. No generalized forces due to gravity exist for the rotational degrees of freedom since the centers of gravity are used as references.

The inertia forces, F_I, are generalized forces and need no further definition. The resulting equations of motion are

$$[M_{i,j}] \{\ddot{g}\} = \{GF_s\} + \{GF_t\} + \{GF_a\} + \{GF_g\} - \{F_I\} \tag{44}$$

The right-hand side of Eq. 44 contains no second derivatives; hence, knowing displacements and velocities, the right-hand side may be evaluated. Multiplying the right-hand side by the inverse of the generalized mass matrix then gives the generalized accelerations. Note that the generalized mass matrix is constant, and may thus be inverted once and for all. Having the generalized accelerations, the generalized displacements and velocities a short time later may be estimated. The process then is repeated, advancing time until the desired time period has been covered. Estimation of displacements and velocities is accomplished using fifth-order open Adams integration.

SECTION 5

COMPUTER PROGRAM

5.1 General Description

A computer program has been prepared based upon the formulations in the previous sections. This program (TRUCK, version 2.0) is written in standard ANSI FORTRAN and consists of 43 user-supplied routines and approximately 4400 card images. This section describes the organization of the code (subsection 5.2), the operation of the code (5.3), program input (5.4), program output (5.5), and presents an example problem in subsection 5.6. In addition, Appendix A presents a macro cross-reference list of all program variables and Appendix B a complete source listing of the code.

As an introduction to the inner workings of the code, the major program steps are outlined below in sequential order:

1) Input data are read. The input is described in detail in division 5.4.

2) The vehicle center of gravity position is calculated and the reference point is shifted to the center of gravity.

3) The λ matrix, which transforms generalized coordinate displacements into spring elongations, is formed.

4) The vehicle is trimmed; that is, the generalized coordinate displacements which place the vehicle in equilibrium under the action of gravity, tire forces, and spring forces are determined. Because the spring and tire forces are non-linear, a trial and error procedure must be used. Basically, each generalized coordinate is perturbed very slightly, and the resulting changes in generalized accelerations are found. This gives approximate local derivatives which can be extrapolated to find an estimated equilibrium state. The process is repeated until acceptable convergence is obtained.

If guy wires are present on the shelter, a preliminary trimming takes place, in which the shelter and racks are trimmed with the guy wire loads at the specified values. This trim takes place with the truck body assumed to be horizontal. This pretrim allows the guy wire force-deflection curves to be biased so that the subsequent overall trim can take place with the guy wires treated as ordinary springs. The overall trim will change the guy wire loads due to the fact that the truck body will not, in general, end up horizontal. The change will be slight, however.

5) The generalized mass matrix is calculated and inverted.

6) The main response loop begins. Within this loop, the following steps take place.

a) The Euler angles are calculated.

b) The β transformation matrix is determined.

c) The earth-fixed axis position of the vehicle is found.

d) The springs are checked to see if any of them have bottomed out. If they have, rebound calculations are made to determine the new generalized velocities, and the numerical integration is restarted.

e) Generalized gravitational forces are calculated.

f) Generalized spring and damper forces are determined.

g) Generalized tire forces are calculated. This calculation requires determining the state of the tire - off the ground, sliding, or not sliding.

h) The generalized aerodynamic forces are determined.

i) The inertia forces are claculated.

j) The above generalized forces are accumulated and the result multiplied by the inverse of the mass matrix to yield the generalized accelerations.

k) Integration takes place to estimate the generalized displacements and velocities one time step forward.

l) The cycle is repeated until the time reaches the stop time specified in the input.

7) Upon completion of the time-history response, a determination is made whether the vehicle has overturned. If the user has selected the iterative option, the program then automatically adjusts the peak incident overpressure and reruns the response (Step 6) until the threshold of overturning is determined to within approximately 2% on overpressure.

8) At this point the process is repeated (Steps 6 and 7) for other blast orientations and yields, if requested, in order to generate vulnerability envelopes or other response runs.

9) A different vehicle can be analyzed, beginning with Step 1.

5.2 Description of Routines, Common Blocks, and Dimensioned Variables

Table 7 lists all the routines and common blocks which make up the TRUCK code and gives the length of each common. An "X" indicates that a particular common belongs in a particular routine. The names of variables within common blocks are the same throughout the program.

Program flow diagrams of the major routines are presented in Figures 9 through 13. These figures indicate in more detail the program steps outlined in division 5.1.

Tables 8, 9, and 10 are intended to provide all the information necessary to increase or decrease dimensions within the program. Table 8 lists in both parametric and numeric forms the current maximum dimensions. These dimensions should also be consulted when forming input (division 5.4) to be sure the data is within the limits of the program.

If either larger or smaller dimensions are desired, Table 9 lists program changes required by specific changes in dimensions. Table 10 then lists all the dimensioned variables which might need to be changed. These variables are all located in common. Variables whose dimensions need not be changed are not listed.

5.3 Program Operation

The TRUCK program was written in ANSI standard FORTRAN IV and developed on a (CDC) 6600 computer under the SCOPE 3.4.4 operating system. The code should be easily adaptable to any modern, large scientific computer. Double precision is required for systems utilizing a relatively short word length.

Blank common is utilized in order to minimize the core required to load and execute. Using compile option "OPT=1" (a medium optimization level) the code requires 29 seconds to compile and 131,000 octal cells of memory to load and execute.

Computation times will vary considerably, depending on the vehicle, the complexity of the model, and the number of orientations considered. The example problem described in subsection 5.6 required 220 cp seconds for 2540 integration steps (primary Δt = 0.5 msec).

Input and output are equated with logical files TAPE5 and TAPE6, respectively. TAPE8 is reserved for graphics, although the code does not include any graphics at this time.

TABLE 7

TRUCK ROUTINES AND COMMON BLOCKS

COMMON	BLANK (LAMB)	ADAM	BLAST	BOTM	CALC	CG	CHECK	CINTI	DATAIN	DELTAS	DIMENS	EULC	EXTDIM	FSPDIM	INTFLG	ITER	JFTRIM	LOAD	MASS	MATDIM	MAXMIN	MOVING	OPTION	PRELOD	RLP	SPRING	TIREC	TIRSAV	TRMDIM	WEIGHT
LENGTH OF COMMON	4400	807	28	400	318	3	1	2	1404	100	4	6	200	200	1	53	1	5488	2500	100	1711	19	75	4	2850	1305	41	324	50	14
MAIN	X		X	X	X	X	X	X	X	X	X				X	X	X	X	X		X	X	X	X		X	X	X		X
ADAMI		X																					X							
ADAM5		X																					X							
AERO			X		X													X					X							
ATMOS																														
BASE	X				X					X													X	X		X				
BETAIJ					X																									
BLOCK DATA									X		X							X								X				
BOTTOM	X	X		X	X					X	X	X			X								X			X				
CGCALC						X			X														X							
DIFFL			X															X												
DRAGL																														
ENGULF			X															X												
EULER					X							X																		
EXTRNL	X	X			X				X				X		X							X	X							
FORCE1					X				X														X							
FSPRNG	X				X		X		X	X	X			X									X			X				
GEOM					X	X	X		X		X												X	X			X			X
GRAV																							X							X
HYTRUK			X																											
INLOAD			X			X												X												
INPSPR				X							X												X			X	X			
INT1								X																						
INT1X																														
LAMBDA	X					X			X														X				X			
MASSM					X				X		X								X				X							
MATXIN																				X										
MAXI					X				X	X											X	X	X							
MOTION		X	X		X	X	X		X	X	X	X			X			X	X		X	X	X							
MULT																														
ORIGIN					X																	X								
PITER																														
PRETR	X				X		X																X	X		X				X
REBOUN	X	X			X				X		X	X			X				X				X							
RELAX																									X					
RFORCE					X				X														X							
SETA			X															X												
SFORCE					X				X														X							
SOLVE																														
SUMARY					X																X		X							
SUMTAB			X		X																X		X							
TIRES		X			X				X						X		X					X	X			X	X	X		
TRIM	X			X	X		X		X	X							X					X	X			X	X		X	X

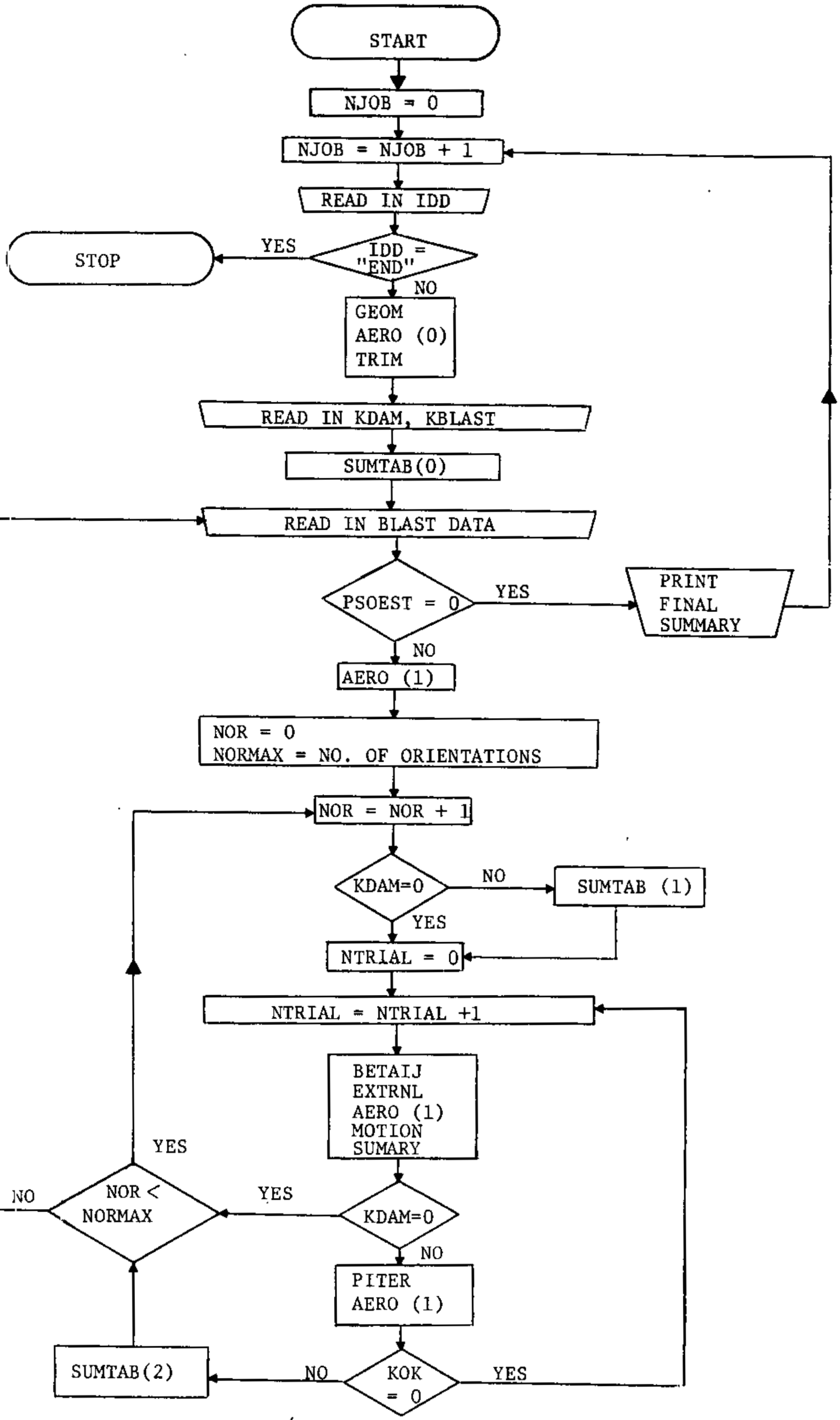

FIGURE 9. PROGRAM TRUCK FLOW DIAGRAM

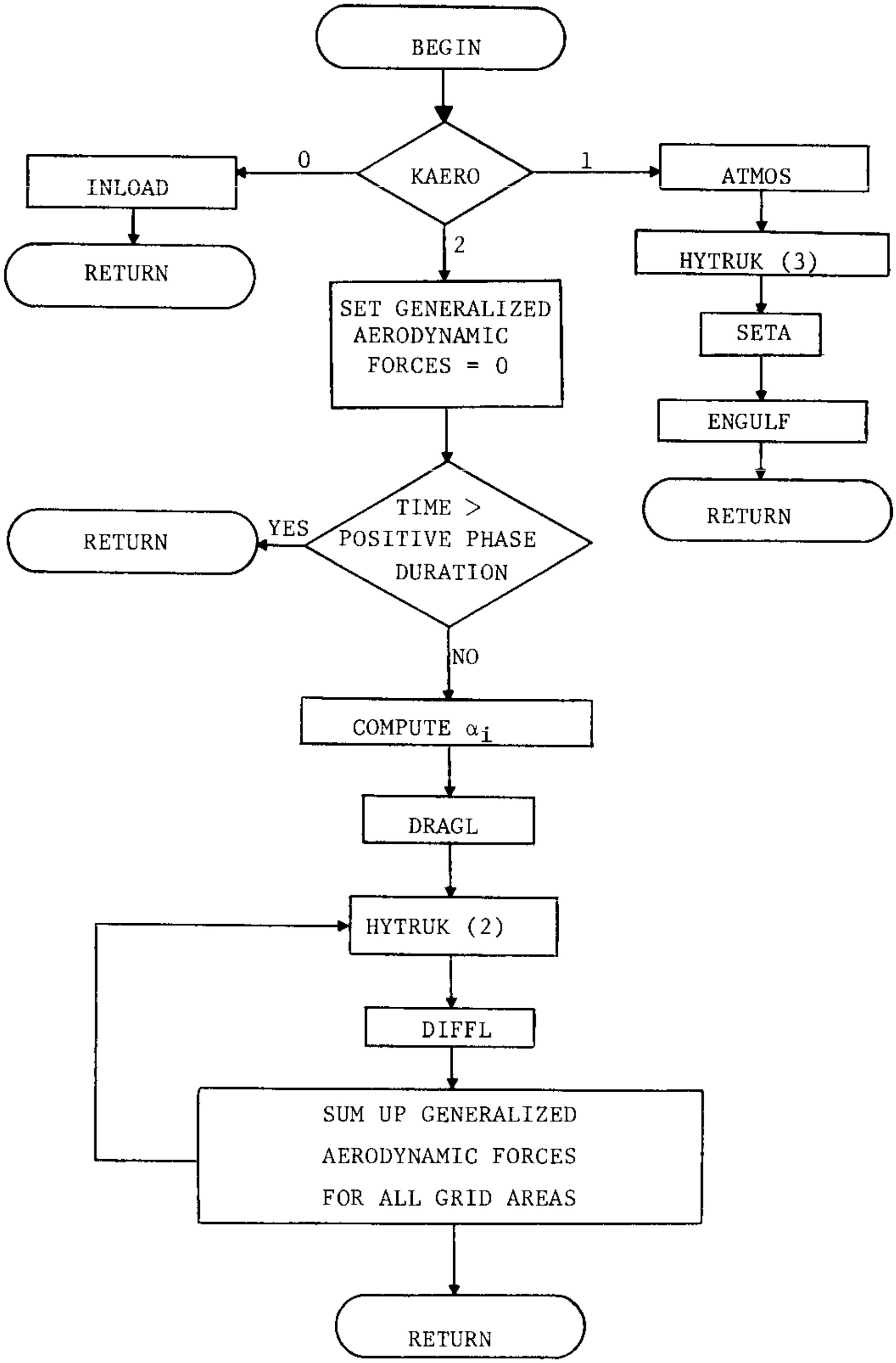

FIGURE 10. FLOW DIAGRAM FOR SUBROUTINE AERO

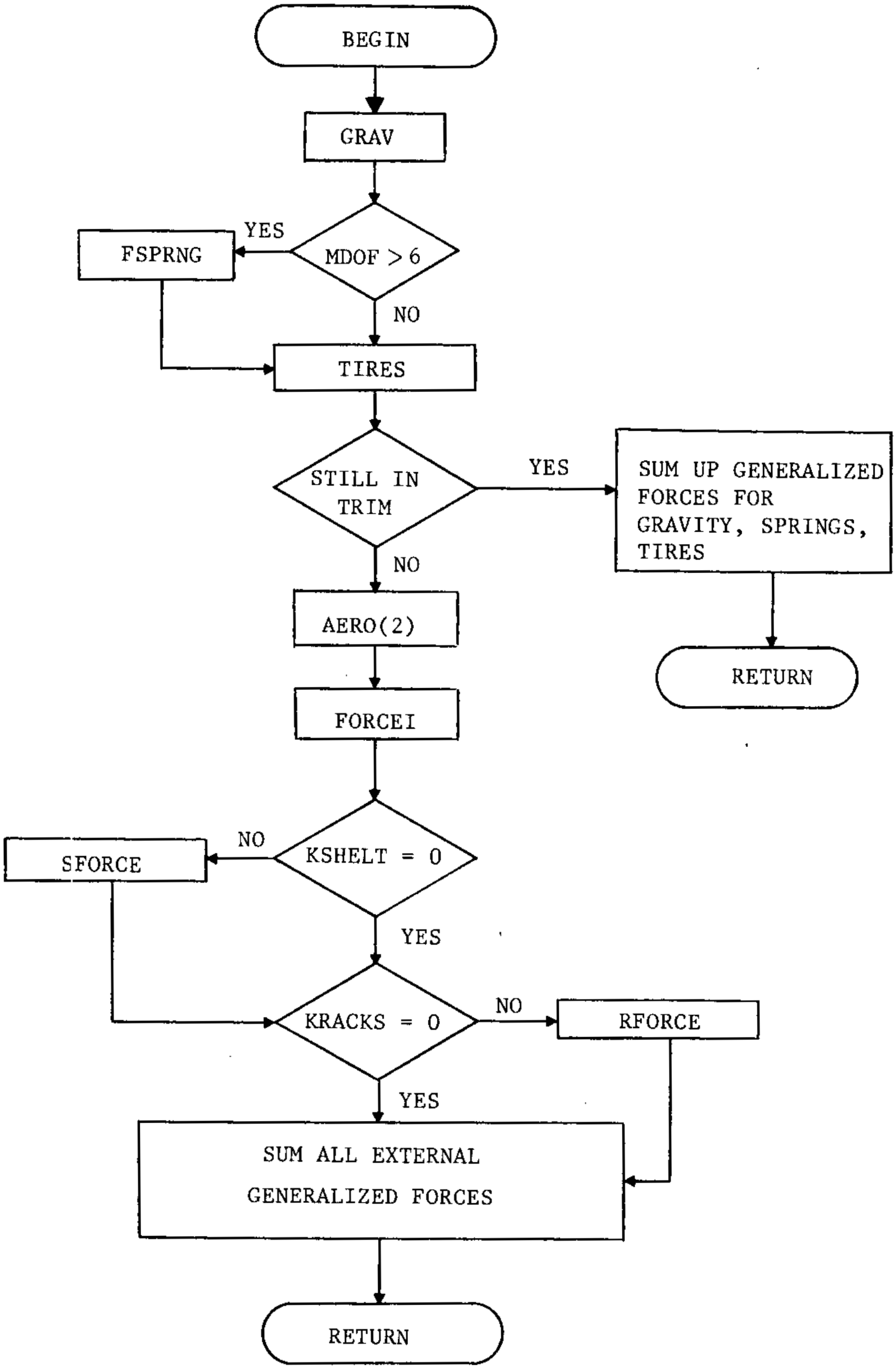

FIGURE 11. FLOW DIAGRAM FOR SUBROUTINE EXTRNL

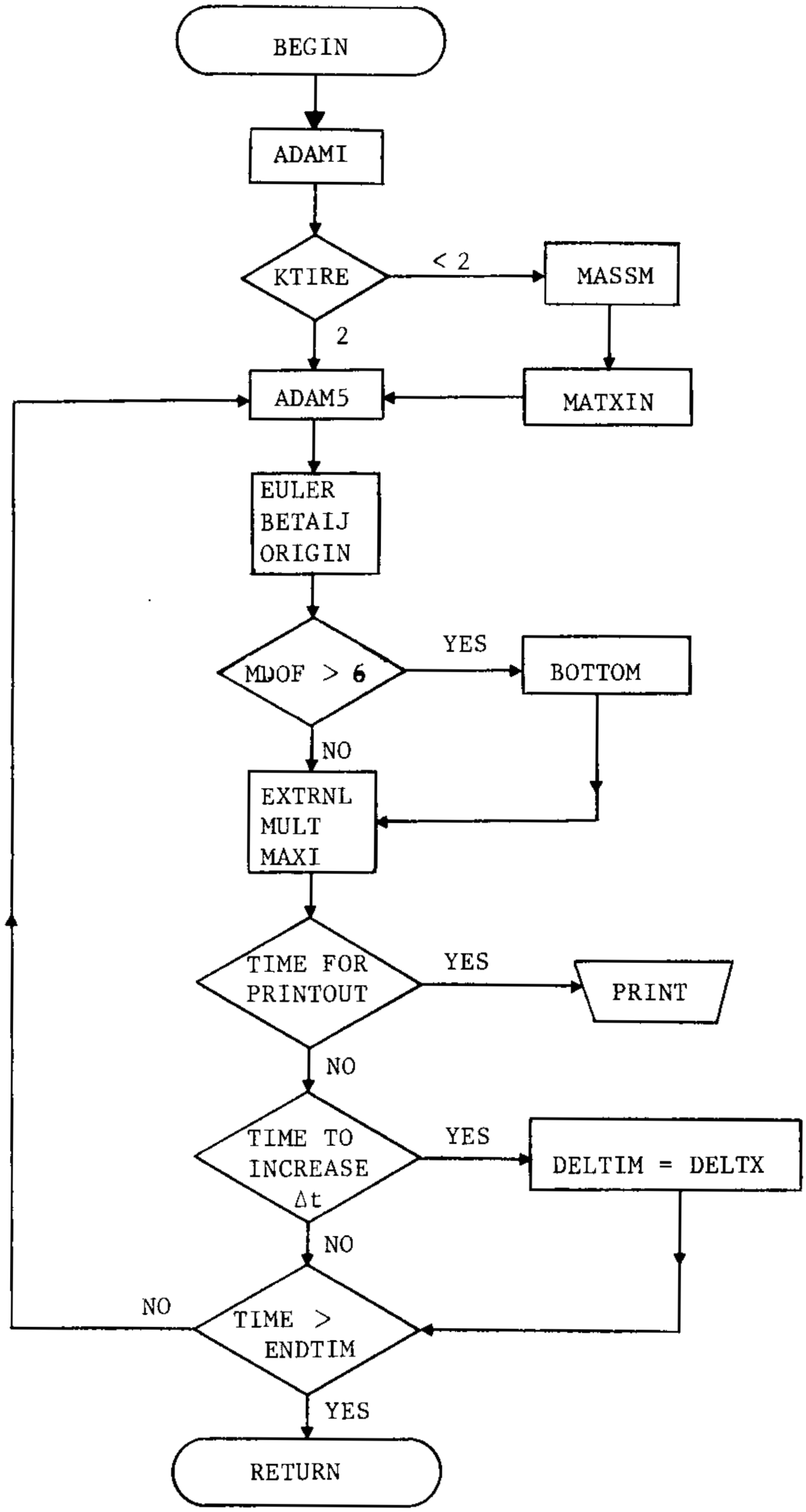

FIGURE 12. FLOW DIAGRAM FOR SUBROUTINE MOTION

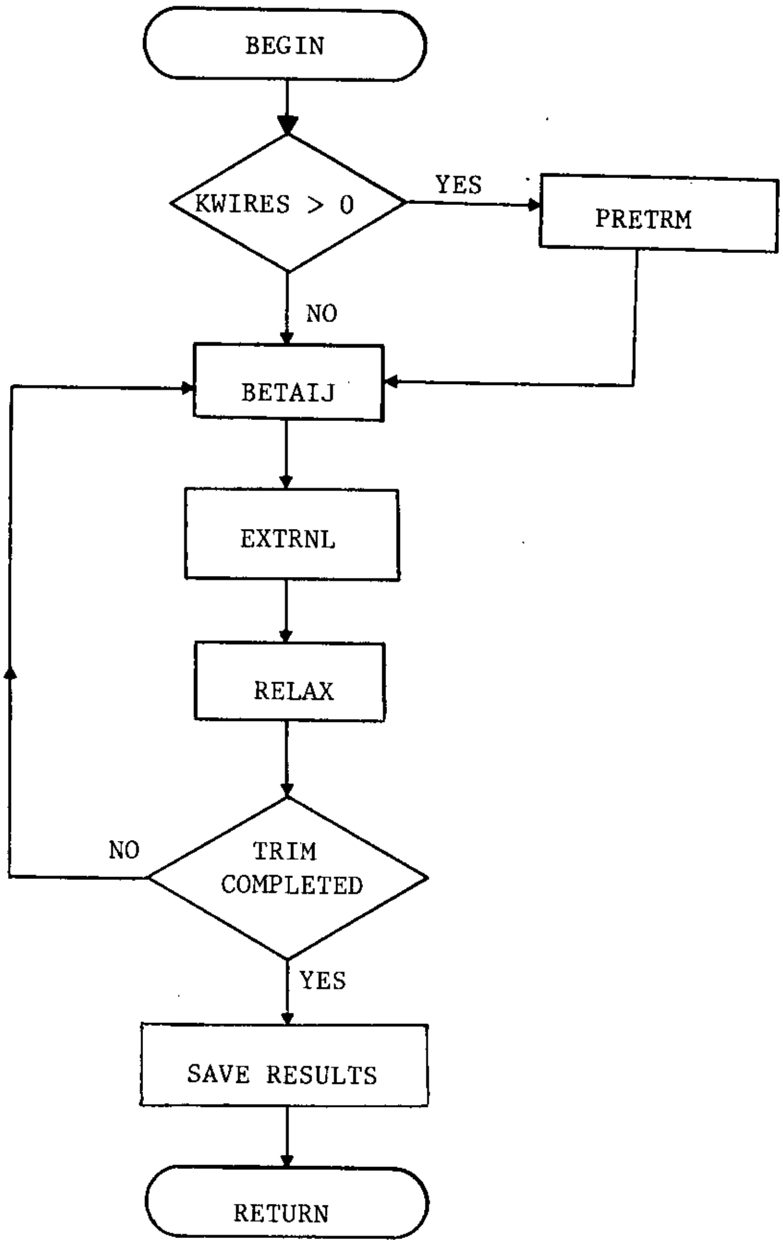

FIGURE 13. FLOW DIAGRAM FOR SUBROUTINE TRIM

TABLE 8

MAXIMUM DIMENSIONS IN TRUCK

Description of Variable	Variable	Maximum Dimension
Number of Shelter Springs per Side	KSHELT	No Limit*
Number of Guy Wires per Side	KWIRES	4*
Number of Racks per Side	KRACKS	3*
Number of Axles	KAXLES	6*
Number of Bogie Springs per Side	KBOG	2*
Number of Tires per Axle per Side	NTIRES	3*
Number of Springs per Axle per Side	NASPR	No Limit*
Number of Springs per Rack per Side	NRSPR	No Limit*
Total Entries in all Force-Velocity Tables per Side	MDIMDP	600
Total Entries in all Force-Displacement Tables per Side	MAXSPR	600
Number of Aerodynamic Boxes	NBOX	4
Number of Aerodynamic Grid Areas per Surface per Box	NPS	16
Number of Distinct Orientations per Job	JMAX	19
Number of Distinct Weapon Yields per Job	KMAX	6
Total Number of Springs	MSPRNG	100*
Total Number of Masses	MASSES	10*
Total Number of Degrees of Freedom	MDOF	50*

*Special Constraints:

$$\text{MSPRNG} \equiv 2\ [(1-\text{KRIGID})(\text{KSHELT}+\text{KWIRES}+\text{KBOG}) + \text{KBOG} + \sum_{\text{KRACKS}} \text{NRSPR} + \sum_{\text{KAXLES}} \text{NASPR}]$$

$$\text{MASSES} \equiv 1 + \text{KRACKS} + \text{KAXLES} - \text{KRIGID} + \begin{cases} 0, & \text{KSHELT}=0 \\ 1, & \text{KSHELT}>0 \end{cases}$$

$$\text{MDOF} \equiv 6 - 5\ \text{KRIGID} + 2\ \text{KAXLES} + 12\ \text{KRACKS} + \begin{cases} 0, & \text{KSHELT}=0 \\ 4, & \text{KSHELT}>0 \end{cases}$$

TABLE 9

PROGRAM CHANGES NECESSITATED BY DIMENSION CHANGES

When Changing the Dimension Corresponding to:	Also Change the Fixed-Point Number in the Indicated Statement	
	Subroutine	Location*
MSPRNG	BLOCK DATA	
MAXSPR	BLOCK DATA	
MDIMDP	BLOCK DATA	
MDOF	BLOCK DATA	
MASSES	BLOCK DATA	
KRACKS	BLOCK DATA	
JMAX	SUMTAB	50^{+4}
KMAX	SUMTAB	50^{+3}
JMAX or KMAX	SUMTAB	50^{-1}
MDOF	MATXIN	10^{-20}
MDOF	RELAX	50^{-4}

*The location code is interpreted as follows:

S^{+n} refers to the n^{th} line after statement number S.

TABLE 10

LIST OF DIMENSIONED VARIABLES

VARIABLE (DIMENSION)	COMMON BLOCK ASSIGNMENT
QR (2*MDOF)	ADAM
QRD (6,2*MDOF)	ADAM
QRP (2*MDOF)	ADAM
EXPAN (MSPRNG)	BOTM
NBOTTOM (MSPRNG)	BOTM
SBOTM (MSPRNG)	BOTM
VEL (MSPRNG)	BOTM
FAERO (MDOF$^{(3)}$)	CALC
FORCE (MDOF)	CALC
GENACC (MDOF)	CALC
GENDIS (MDOF$^{(2)}$)	CALC
GENVEL (MDOF$^{(2)}$)	CALC
XBARCM (3, MASSES+KRACKS)	CALC
CGMASS (MASSES+KRACKS)	DATAIN
CGPOS (3, MASSES+KRACKS)	DATAIN
DAMPF (MDIMDP)	DATAIN
DAMPV (MDIMDP)	DATAIN
ISDAMP (MSPRNG/2)	DATAIN
NPDAMP (MSPRNG/2)	DATAIN
XI1 (MASSES+KRACKS)	DATAIN
XI2 (MASSES+KRACKS)	DATAIN
XI3 (MASSES+KRACKS)	DATAIN
DISP (MSPRNG)	DELTAS

TABLE 10 (CONT'D)

LIST OF DIMENSIONED VARIABLES

VARIABLE (DIMENSION)	COMMON BLOCK ASSIGNMENT
FGRAV (MDOF)	EXTDIM
FI (MDOF)	EXTDIM
FTIRES (MDOF)	EXTDIM
SPRNG (MDOF)	EXTDIM
FORC (MSPRNG)	FSPDIM
V (MSPRNG)	FSPDIM
GENDIX (MDOF)	ITER
XLAMDA (MSPRNG, MDOF-6)	BLANK COMMON
JFIR (NPS, 6, NBOX)	LOAD
KTS (NBOX)	LOAD
NPS (6, NBOX)	LOAD
S (NPS, 6, NBOX)	LOAD
SI1 (NPS, 6, NBOX)	LOAD
SI2 (NPS, 6, NBOX)	LOAD
SI3 (NPS, 6, NBOX)	LOAD
SI4 (NPS, 6, NBOX)	LOAD
SJ1 (NPS, 6, NBOX)	LOAD
SJ2 (NPS, 6, NBOX)	LOAD
SJ3 (NPS, 6, NBOX)	LOAD
SJ4 (NPS, 6, NBOX)	LOAD
TD (NPS, 6, NBOX)	LOAD

TABLE 10 (CONT'D)

LIST OF DIMENSIONED VARIABLES

VARIABLE (DIMENSION)	COMMON BLOCK ASSIGNMENT
XBOX (NPS, 6, NBOX)	LOAD
YBOX (NPS, 6, NBOX)	LOAD
ZBOX (NPS, 6, NBOX)	LOAD
XMASS (MDOF[(1)], MDOF[(1)])	MASS
KOL(MDOF)	MATDIM
ROW(MDOF)	MATDIM
AMAX (MDOF)	MAXMIN
AMIN (MDOF)	MAXMIN
ANGLE (JMAX)	MAXMIN
CIP (JMAX, KMAX)	MAXMIN
CIQ (JMAX, KMAX)	MAXMIN
CPSIG (JMAX, KMAX)	MAXMIN
CPSO (JMAX, KMAX)	MAXMIN
CRKM (JMAX, KMAX)	MAXMIN
DMAX (MDOF)	MAXMIN
DMIN (MDOF)	MAXMIN
IFL (JMAX, KMAX)	MAXMIN
SMAX (MSPRNG)	MAXMIN
SMIN (MSPRNG)	MAXMIN
TAMAX (MDOF)	MAXMIN
TAMIN (MDOF)	MAXMIN
TDMAX (MDOF)	MAXMIN

TABLE 10 (CONT'D)

LIST OF DIMENSIONED VARIABLES

VARIABLE (DIMENSION)	COMMON BLOCK ASSIGNMENT
TDMIN (MDOF)	MAXMIN
TSMAX (MSPRNG)	MAXMIN
TSMIN (MSPRNG)	MAXMIN
TVMAX (MDOF)	MAXMIN
TVMIN (MDOF)	MAXMIN
VMAX (MDOF)	MAXMIN
VMIN (MDOF)	MAXMIN
YIELD (KMAX)	MAXMIN
KAXLB (KBOG)	OPTION
NASPR (KAXLES)	OPTION
NRSPR (KRACKS)	OPTION
XBOGIE (KBOG)	OPTION
PRELOD (KWIRES)	PRELOD
DELX (MDOF)	RLP
IP (MDOF)	RLP
PRES (MDOF)	RLP
PX (MDOF)	RLP
RRES (MDOF)	RLP
SIGX (MDOF)	RLP
XRES (MDOF, MDOF)	RLP
XX1 (MDOF)	RLP

TABLE 10 (CONCL'D)

LIST OF DIMENSIONED VARIABLES

VARIABLE (DIMENSION)	COMMON BLOCK ASSIGNMENT
FOFX (MAXSPR)	SPRING
JCURVE (MSPRNG/2 + 2)	SPRING
NPPC (MSPRNG/2 + 2)	SPRING
X (MAXSPR)	SPRING
NTIRES (KAXLES)	TIREC
TLX (NTIRES, KAXLES)	TIREC
TLY (KAXLES)	TIREC
TLZ (KAXLES)	TIREC
DELTNX(3, 2, NTIRES*KAXLES)	TIREC
DELTN1 (3, 2, NTIRES*KAXLES)	TIREC
DTTNX (2, NTIRES*KAXLES)	TIREC
DTTN1 (2, NTIRES*KAXLES)	TIREC
SLID (2, NTIRES*KAXLES)	TIREC
ER (MDOF[(2)])	TRMDIM
WGTS (MASSES+KRACKS)	WEIGHT

Special Constraints:

(1) Dimension must be MDOF or 6, whichever is greater.

(2) Dimension must be MDOF or 8, whichever is greater.

(3) Dimension must be MDOF or 12, whichever is greater.

5.4 Input Data

This section is intended to provide the user with the necessary instructions to generate card input for the TRUCK code. Tables 11 and 12 contain the basic input instructions, while the following discussion contains general remarks and amplification of some of the specific instructions.

The input data are specified in groups, where each group begins on a separate card. More than one card may be required for a group, however. The variable type and format corresponding to each data group are given in Table 12 and are always in fields of 12, except for Group 1. For convenience, floating point numbers can be left justified in the field as long as the exponent is right justified. Also, zero values can be replaced by a blank field. Columns 73 through 80 are not used for data and can be used for card identification or other purposes.

All input parameters, where appropriate, should be compared with the maximum dimensions provided for in the program, as delineated in Table 8. This is important since the program does not check all input for violations.

In general terms, the input consists of defining: 1) the mechanical model of the vehicle, 2) the aerodynamic model, and 3) the blast characteristics. The axis system used to define both the vehicle and aerodynamic models is shown in Figure 1. The origin to be used for data input is arbitrary except that the x = 0 plane defines the vehicle plane of symmetry. The units to be used are as follows:

Mass in $lb\text{-}sec^2/in$
Length in inches (unless otherwise stated)
Time in seconds

The restriction to the above set of units results from the fact that the blast characteristics routines provide pressure in lbs/in^2. Note that the acceleration due to gravity in Group 23 of Table 12 must be specified as 386.0 in order to be consistent with the above units.

Group 1 of the data represents a free field run identifier which might contain vehicle identification, the date, etc. It is not used within the program, unless it begins with "END". The "END" represents a flag which terminates the run; otherwise multiple runs can be stacked one behind the other.

Group 2 dictates whether the vehicle has conventional tires (IOPT=0) or is a tracked vehicle such as a tank (IOPT=1).

The basic components of the vehicle system are indicated in Group 3. Table 11 should be consulted to make sure a consistent set of data is selected.

TABLE 11

ACCEPTABLE INPUT PARAMETERS FOR
MAJOR PROGRAM OPTIONS

INPUT PARAMETER	OPTION A RIGID BODY - THREE DEGREES OF FREEDOM	OPTION B MINIMUM OF 10 DEGREES FREEDOM
KRIGID	1	0
KAXLES	1	2,3,4,5,6
KSHELT	0	0,1,2,.....
KWIRES	0	0 if KSHELT=0; 0,1,2,3,4 otherwise
KBOG	0	0, 1, 2
KRACKS	0	0, 1, 2, 3
KTS (N) (all N)	1	1, 2

TABLE 12

INPUT DATA

GROUP	FORMAT	INPUT DATA
1	40A2	Run identification, up to 80 characters: IDD
2	I12	Vehicle type: IOPT 0, Wheeled Vehicle 1, Tracked Vehicle
3	6I12	Configuration data: KSHELT, KWIRES, KRACKS, KAXLES, KBOG, KRIGID KSHELT = Number of springs (not including guy wires) attaching left side of shelter to left side of truck (0 if shelter is rigidly attached to truck or no shelter at all). KWIRES = Number of guy wires on left side of shelter (0 if KSHELT = 0). KRACKS = Number of racks in left side of shelter (0 if racks are rigidly attached to shelter). KAXLES = Number of axles on truck. KBOG = Number of bogie springs on left side of truck. KRIGID = Rigid body flag: 0 gives ordinary run; 1 gives three degrees of freedom run - roll, sideslip and heave. See Table 11.
4	6I12	Number of tires on left end of each axle: NTIRES (I), I=1, KAXLES
5	6I12	Number of springs on left side of each axle, not including bogie springs: NASPR (I), I=1, KAXLES (0 if KRIGID=1)
6	6I12	Number of springs on each rack in left side of shelter: NRSPR(I), I=1, KRACKS (Omit Group 6 if KRACKS = 0)

TABLE 12 (CONT'D)

INPUT DATA

GROUP	FORMAT	INPUT DATA
7	4F12.1	Masses and their locations. Each card contains the mass value and the x, y, and z coordinates of its center of gravity. One card for each mass. Order of masses is: Vehicle body Axles (front to rear, include in vehicle body mass if KRIGID=1) Shelter (included in vehicle body mass if shelter rigidly attached) Racks in left side of shelter (<u>all</u> racks included in vehicle body mass if racks rigidly attached). The number of cards will be (1+KAXLES+KRACKS) if KSHELT=0, and (2+KAXLES+KRACKS) if KSHELT>0. If KRIGID=1, there is only one card.
8	F12.1	Product of inertia of mass 1, I_{yz}. Other products of inertia are assumed to be zero.
9	3F12.1	Mass moments of inertia, I_{xx}, I_{yy}, I_{zz}. The number of cards is the same as in group 7.
10	6F12.1	Attachment points for axle springs. Each card will contain the x, y, z coordinates of the point on the vehicle body to which the spring is attached and the x, y, z coordinates of the point on the axle to which the spring is attached. One card for each left-side axle spring. The number of cards will be $\sum_{I=1}^{KAXLES} NASPR(I)$. (omit Group 10 if KRIGID=1)
11	2F12.1	y and z - coordinates of one axle.
12	6F12.1	x - coordinates for wheels on axle described in Group 11, ordered from inside-out. (Repeat Groups 11 and 12 for each axle)

TABLE 12 (CONT'D)

INPUT DATA

GROUP	FORMAT	INPUT DATA
13	I12,F12.1	Attachment points for bogie springs. One card for each left-side bogie spring. Each card will contain the number of the forward axle of the pair comprising the bogie and the x coordinate of the bogie spring. (The bogie spring is assumed to act parallel to the y axis and the bogie is assumed to be symmetrical.) (Omit Group 13 if KBOG=0)
14	6F12.1	Attachment points for shelter springs. One card for each left-side shelter spring. Each card will contian the x, y, z coordinates of the point on the vehicle body to which the spring is attached and the x, y, z coordinates of the point on the shelter to which the spring is attached. (Omit Group 14 if KSHELT=0).
15	6F12.1	Attachment points for guy wires. One card for each left-side guy wire. Each card will contain the x, y, z coordinates of the point on the vehicle body to which the guy wire is attached and the x, y, z coordinates of the point on the shelter to which the guy wire is attached. (Omit Group 15 if KWIRES=0).
16	6F12.1	Attachment points for rack springs. Each card will contain the x, y, z coordinates of the point on the left rack to which the spring is attached and the x, y, z coordinates of the point on the shelter to which the spring is attached. One card for each left-side rack spring. The number of cards will be $\sum_{I=1}^{KRACKS} NRSPR(I)$. (Omit Group 16 if KRACKS=0).
17	6F12.1	Preloads in guy wires, one number for each left-side guy wire. (Omit Group 17 if KWIRES=0).

TABLE 12 (CONT'D)

INPUT DATA

GROUP	FORMAT	INPUT DATA
18	I12,(6F12.1)	Non-linear damping force-velocity curves. One curve for each left-side spring. Order of the springs is: Axle springs on left side Bogie springs on left side Shelter springs on left side Guy wires on left side Rack springs on left side The number of curves will be $\sum_{I=1}^{KAXLES} NASPR(I)+KBOG+KSHELT+KWIRES+\sum_{I=1}^{KRACKS} NRSPR(I)$ (If KRIGID=1, there will be no curves, so this group will be omitted). Each curve will be described by several points on the curve. The points must proceed from the compression stroke (negative velocity, positive force) to the rebound stroke (positive velocity, negative force). The extreme compression and rebound velocities must be large enough to include any velocities which may be encountered during the response calculations. For each curve, the first card will contain the number of points that will be specified for the curve. Subsequent cards will contain velocity-force pairs, each pair defining a point. There will be three pairs per card. The program dimensions permit the number of curves to be MSPRNG/2-KBOG. The total number of points defining the curves can be no greater than MDIMDP (see Table 8).
19	2F12.1	Tire damping coefficients. Forces per unit velocity in the radial and tangential directions. Both coefficients should be input as positive numbers.
20	I12,(6F12.1)	Non-linear spring force-displacement curves. One curve for each left-side spring. The order and number are as described in Group 18. Each curve will be described by several points on the curve. The points must proceed from compression (negative displacement, positive force) to tension (positive

TABLE 12 (CONT'D)

INPUT DATA

GROUP	FORMAT	INPUT DATA
		displacement, negative force). The extreme compression and tension displacements are interpreted as defining bottoming of the spring. For each curve, the first card will contain the number of points that will be specified on the curve. Subsequent cards will contain displacement-force pairs, each pair defining a point. There will be three pairs per card. The total number of points defining the curves, including groups 21 and 22, can be no greater than MAXSPR (see Table 8).
21	I12,(6F12.1)	Tire non-linear force-normal displacement data. Specified in same manner as Group 20 data, except that positive displacements will, of course, give zero force and bottoming out is not included. Hence, first data point should be a large enough (negative) displacement to include any displacement which may be encountered during the response calculations and last point specified should be 0.0, 0.0.
22	I12,(6F12.1)	Tire non-linear force-tangential displacement data. Specified in same manner as Group 20 data, except that only magnitudes are needed and there is no bottoming out. Hence, first data point specified should be 0.0, 0.0, and last data point should be a large enough displacement to include any displacement which may be encountered during the response calculations. All displacements and forces should be positive.
23	3F12.1	Tire radius, coefficient of sliding friction between tire and ground, and acceleration due to gravity.
24	3F12.1	Ground slope data. Ground slopes in regions 1 and 2, in radians, positive if height increases as one moves in the positive x direction; and the x coordinate at which the slope changes (see Figure 2).

TABLE 12 (CONT'D)

INPUT DATA

GROUP	FORMAT	INPUT DATA
25	2F12.1	Response time at which run ends and primary integration time step.
26	I12	Number of primary time steps between print outs.
27	I12	Number of rectangular boxes used in aerodynamic model - includes both truck and shelter boxes, NBOX.
28	4I12	Code designating type of each box: KTS(N), N=1, NBOX 1 - Box (shelter or otherwise) rigidly attached to truck 2 - Box represents shelter and is not rigidly attached to truck.
29	I12	Number of aerodynamic load points on surface, NPS(I,N), where I refers to the surface code (see Figure 14). If no load is applied set NPS = 0. (Note: If NPS (I,N)=0, skip groups 30 and 31.)
30	4F12.1	Loading information: XB(K,I,N) = x-position of load point. YB(K,I,N) = y-position of load point. ZB(K,I,N) = z-position of load point. S(K,I,N) = Area associated with load point.
31	4F12.1	Loading information: SI1(K,I,N) = s_1 SI2(K,I,N) = s_2 SI3(K,I,N) = s_3 SI4(K,I,N) = s_4 (See Figures 4 and 8 for definition of s. Data for I = 5 and 6 are not required because diffraction loading is not used on the top and bottom surfaces. Therefore, blank cards should be used.) Repeat groups 30 and 31 for K=1, NPS(I,N) Repeat groups 29, 30, and 31 for I=1, 3, 4, 5, and 6.* Repeat groups 29, 30, and 31 for N=1, NBOX

TABLE 12 (CONT'D)

INPUT DATA

GROUP	FORMAT	INPUT DATA
		*Data for I=2 is omitted because of symmetry with I=1.
32	2I12	Code for controlling iteration: KDAM. 0, No iteration 1, Iterate on overpressure to determine overturning condition.
		Blast model code: KBLAST. 1, 1 KT sea level model 2, 60 $W^{1/3}$ meters height of burst model
33	6F12.1	Blast data: PSO, AD1, AD2, DELTA, W, ALT. PSO - Estimate of incident shock overpressure, psi AD1 - Azimuthal angle 1, degrees ($-90 \leq AD1 \leq 90$) AD2 - Azimuthal angle 2, degrees ($-90 \leq AD2 \leq 90$) DELTA - Increment in angle, degrees W - Nuclear weapon yield, KT ALT - Altitude of vehicle above sea level, ft.

NOTE - Program will compute all cases from AD1 to AD2 every DELTA degrees, then look for another set of data under group 33 (see Figure 3). A value of PSO = 0.0 (or a blank card) will terminate group 33. Additional jobs can be processed by returning to GROUP 1. The entire run terminates with an "END", beginning in column 1 of GROUP 1.

If a bogie configuration is used, stops on the bogie pitch motion must be specified. Otherwise, if the wheels are off the ground, the bogie pitch motion will be unlimited since the bogie is completely free to pitch. Stops are provided by placing a spring on each bogie axle. The spring can supply zero force, but the last deflection specified will act as a stop (see Group 20).

The data regulating the numerical integration are contained in Groups 25 and 26. Early in the response the program uses a relatively small time interval of 0.15 msec between steps to capture the rapidly changing diffraction phase of the blast loading. At 30 msec, however, the program switches to the primary Δt, as read in, for the remainder of the response.

Several different Δt's may have to be tried in order to determine the largest Δt which still permits an accurate solution. The total computational time is nearly proportional to the Δt selected, so it is important to optimize the selection. The Δt should be halved or doubled, as the case may be, to determine at what point the solution becomes significantly affected by the choice of time interval. Once a Δt has been selected for a vehicle, it should remain valid for other blast levels and orientations.

The printout frequency pertains to the primary time interval; if zero, all time history output is suppressed. The stop time for the response should be large enough to capture peak response. For an iteration run (Group 32), the response will automatically terminate when the vehicle reaches a point of "no return" as far as overturning is concerned.

The input data for the aerodynamic model begins with Group 27. The vehicle is broken down into an assemblage of rectangular boxes. Each box has, of course, six surfaces as shown in Figure 14. As an example, a truck carrying a shelter might be described using three boxes, one for the shelter, one for the truck cab, and one for the remainder of the truck body. A number of aerodynamic load points may be defined on each surface of each box. A load point is defined by its location, its area, and the appropriate values of s as defined by Figures 4 and 8.

In Group 32 the user selects whether the run is to iterate on overpressure until the point at which the vehicle barely overturns is reached. Typically the program will require 3 or 4 iterations to determine the critical overpressure.

The blast wave characteristics are also selected in Group 32 by either specifying a waveform corresponding to a free-air, sea level burst, or a 60 $W^{1/3}$ meters height of burst model. The characteristics of each are contained within the program.

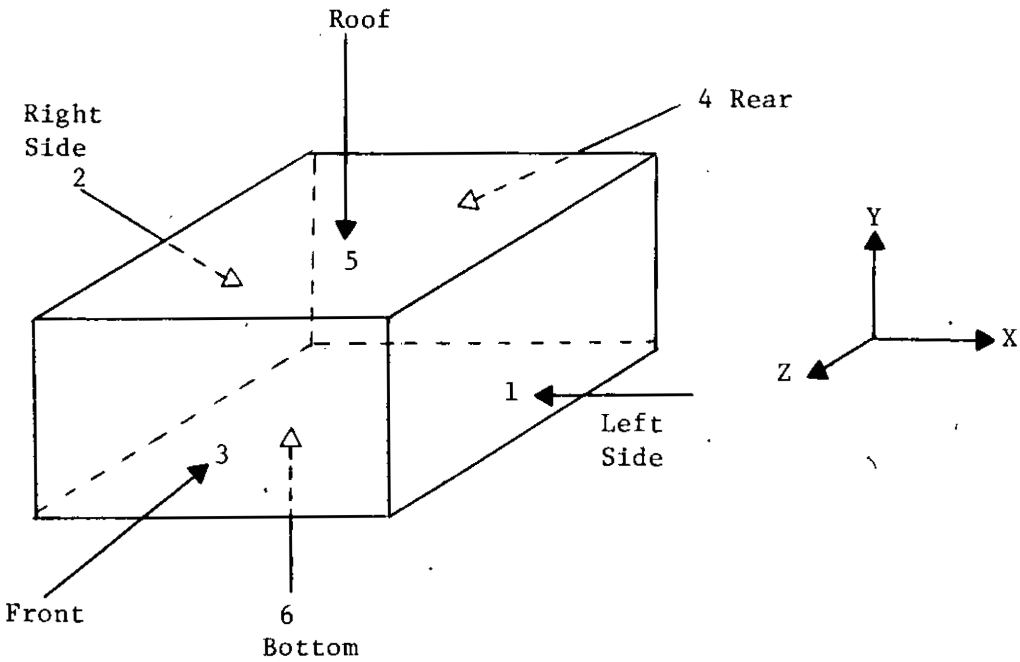

FIGURE 14. DEFINITION OF SURFACE CODES ON AERODYNAMIC BOX

In Group 33 key blast data are specified. An azimuthal angle of 0 degrees corresponds to a (left) side-on exposure; + 90 a head-on exposure. A series of angles can be easily specified by making use of the automatic increment (DELTA). In this case a new estimated overpressure is calculated based on the results of the previous angle. Additionally, the program processes new blast data on subsequent cards for Group 33 until a blank card is reached.

At this point the program prints out a vulnerability summary (for an iteration run) and then returns to Group 1, ready to process another data deck.

5.5 Output

The normal program output consists of three parts; preliminary information, time-history output, and summary information. The preliminary information consists of printing out the input data and certain center of gravity calculations.

The time history output occurs at intervals defined by the analyst in Group 26 of Table 12. The output consists of the time, the generalized coordinate displacements, velocities, and accelerations, the Euler angles, and the center of gravity displacements in the earth-fixed axis system. The order for the generalized coordinates is as defined in Table 1. Also printed is the "CG DISTANCE." This parameter has to do with the vehicle overturning. The program calculates a footprint defined by the projections onto a horizontal plane of the front and rear tire positions. The minimum distance of the vehicle center of gravity from the edge of this footprint is printed out. If the center of gravity moves outside the footprint, the vehicle is overturning.

The summary information includes the maximum and minimum generalized coordinate displacements, velocities, and accelerations and the corresponding times. Maximum and minimum spring deflections and the corresponding times are also printed out. In interpreting these data, it must be noted that the spring numbers do not correspond to the spring number in the input. There are two reasons why these do not correspond. First, only left-side springs are input, so two springs exist for each spring in the input. Thus, in the summary output, odd-numbered springs are left-side springs, even-numbered springs are right-side springs. Secondly, each bogie spring ends up as four springs, one on each side of each axle involved. The actual spring displacement is really the average of the two displacements on the same side. The "spring displacements" for bogey springs should thus be recognized as really being the axle displacements at the bogey spring attachment points.

In addition to the above information, the program also indicates whether the vehicle has overturned, and the time at which it either passed the point of no return, or at which the minimum value of "CG Distance" was recorded for a stable response. For an unstable response the velocity with which the center of gravity crossed the footprint boundary is also printed.

For iteration runs there is one additional summary printed after all orientations and yields have been completed for a particular vehicle. In tabular format the blast parameters corresponding to critical response are collected for each case. It should be noted that when iterating on overpressure the program always makes a final estimate of overpressure after the last response run, so that the final result may actually represent a stable or an unstable response. The important thing is that the estimate is considered close enough to the actual critical overpressure level for all practical applications.

If an error of any kind is detected during a run the program will print out an appropriate message and indicate the subroutine producing the message. If possible the program will either continue with the response or cycle back for the next orientation or yield, or another data deck. An input error will force termination of the run.

5.6 Example Problem

The example problem deals with an M35A2 truck carrying a retro-fitted S280 shelter. This configuration was tested in Operation Dice Throw. The truck is shown in Figure 15. The two rear axles are connected in a bogie arrangement. The shelter is assumed to be rigidly connected to the truck bed and there are two racks in each side of the shelter. The origin for the input data is the center of the front axle. Physical data on the truck are contained in Reference 9. The aerodynamic model is made up from two boxes, one representing the hood and the other the remainder of the truck plus the shelter.

Table 13 contains a listing of the input data. It is annotated with the Group number and other information to assist in correlating the data with the description of the input data in Table 12.

The outputs from the example problem run are illustrated in Tables 14-16. Table 14 contains the preliminary information, Table 15 presents excerpts from the time-history output, and Table 16 contains the summary information.

---

[9] Radkowski, Peter, Letter to Noel Ethridge, BRL, dated 24 June, 1976.

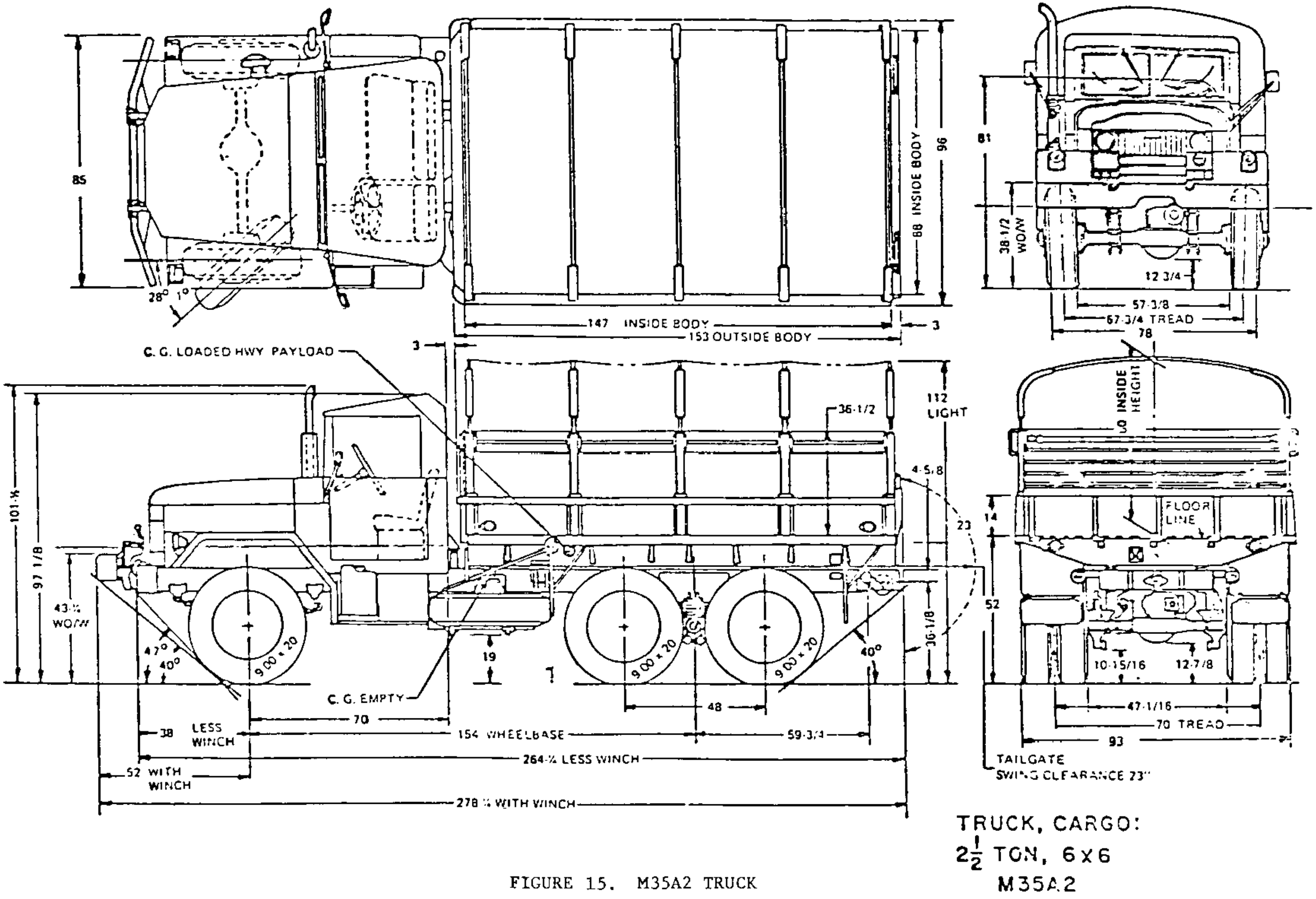

FIGURE 15. M35A2 TRUCK

TABLE 13

SAMPLE PROBLEM INPUT DATA

Sample Data

							Group No.
M35A2 WITH RETROFIT SHELTER RIGIDLY ATTACHED, RACKS SPRING-MOUNTED.							(1)
	0						(2)
	0	0	2	3	1	0	(3)
	1	2	2				(4)
	2	1	1				(5)
	11	11					(6)
28.14	0.0	31.69	-85.77				(7)
3.66	0.0	0.0	0.0				
4.23	0.0	0.0	-130.0				
4.23	0.0	0.0	-178.0				
1.20	29.01	74.51	-126.96				
1.20	29.01	74.51	-173.04				
-19600.							(8)
112400.	97400.	40900.					(9)
0.0	2530.	2530.					
0.0	7730.	7730.					
0.0	7730.	7730.					
547.0	66.6	405.0					
547.0	66.6	405.0					
15.4	10.8	0.0	15.4	5.6	0.0		(10)
15.4	-1.4	-3.24	15.4	11.19	-10.76		
20.24	0.	-130.	20.24	5.	-130.		
20.24	0.	-178.	20.24	5.	-178.		
0.	0.						(11)
33.874							(12)
0.	-130.						(11)
29.25	40.75						(12)
0.	-178.						(11)
29.25	40.75						(12)
	2	20.24					(13)
29.01	41.87	-120.73	35.0	41.87	-120.73		(16)
29.01	41.67	-133.19	35.0	41.87	-133.19		
29.01	102.37	-121.35	35.0	102.37	-121.35		
29.01	102.37	-132.57	35.0	102.37	-132.57		
34.47	41.87	-120.93	34.47	35.0	-120.93		
23.55	41.87	-120.93	23.55	35.0	-120.93		
34.47	41.87	-132.99	34.47	35.0	-132.99		
23.55	41.67	-132.99	23.55	35.0	-132.99		
22.32	41.87	-126.96	22.32	41.87	-120.0		
35.70	41.87	-126.96	35.70	41.87	-120.0		
37.51	110.87	-126.96	37.51	110.87	-120.0		
29.01	41.87	-166.81	35.0	41.87	-166.81		
29.01	41.87	-179.27	35.0	41.87	-179.27		
29.01	102.37	-167.43	35.0	102.37	-167.43		
29.01	102.37	-178.65	35.0	102.37	-178.65		
34.47	41.87	-147.01	34.47	35.0	-147.01		
23.55	41.87	-147.01	23.55	35.0	-147.01		
34.47	41.87	-179.07	34.47	35.0	-179.07		

						Group No.
23.55	41.87	-179.07	23.55	35.0	-179.07	
22.32	41.87	-173.04	22.32	41.87	-160.0	
35.70	41.87	-173.04	35.70	41.87	-160.0	
37.51	110.87	-173.04	37.51	110.87	-160.0	
	5					(18)
-10000.	6540.	-5.0	44.	0.0	0.0	
5.0	-81.	10000.	-18320.			
	5					
-10000.	26160.	-5.0	175.	0.0	0.0	
5.0	-325.	10000.	-73300.			
	2					
-100.	0.	100.	0.			
	2					
-100.	0.	100.	0.			
	5					
-10000.	13080.	-5.0	68.	0.0	0.0	
5.0	-162.	10000.	-36600.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	50700.	10000.	-50700.			
	2					
-10000.	50700.	10000.	-50700.			
	2					
-10000.	76050.	10000.	-76050.			
	2					
-10000.	76050.	10000.	-76050.			
	2					
-10000.	76050.	10000.	-76050.			
	2					
-10000.	76050.	10000.	-76050.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	50700.	10000.	-50700.			
	2					
-10000.	50700.	10000.	-50700.			
	2					
-10000.	76050.	10000.	-76050.			

						Group No.
	2					
-10000.	76050.	10000.	-76050.			
	2					
-10000.	76050.	10000.	-76050.			
	2					
-10000.	76050.	10000.	-76050.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	101400.	10000.	-101400.			
	2					
-10000.	101400.	10000.	-101400.			
5.0	5.0					(19)
	3					(20)
-6.2	6200.	0.	0.	100.	-100000.	
	2					
-100.	0.0	100.	0.0			
	2					
-5.	0.	100.	0.			
	2					
-5.	0.	100.	0.			
	3					
-6.2	7130.	0.	0.	100.	-115000.	
	2					
-0.5	5100.	0.5	-5100.			
	2					
-0.5	5100.	0.5	-5100.			
	2					
-0.5	2550.	0.5	-2550.			
	2					
-0.5	2550.	0.5	-2550.			
	2					
-0.5	3825.	0.5	-3825.			
	2					
-0.5	3825.	0.5	-3825.			
	2					
-0.5	3825.	0.5	-3825.			
	2					
-0.5	3825.	0.5	-3825.			
	2					
-0.5	5100.	0.5	-5100.			
	2					
-0.5	5100.	0.5	-5100.			
	2					
-0.5	5100.	0.5	-5100.			
	2					
-0.5	5100.	0.5	-5100.			
	2					
-0.5	5100.	0.5	-5100.			

								Group No.
	2							
-0.5	2550.	0.5	-2550.					
	2							
-0.5	2550.	0.5	-2550.					
	2							
-0.5	3825.	0.5	-3825.					
	2							
-0.5	3825.	0.5	-3825.					
	2							
-0.5	3825.	0.5	-3825.					
	2							
-0.5	3825.	0.5	-3825.					
	2							
-0.5	5100.	0.5	-5100.					
	2							
-0.5	5100.	0.5	-5100.					
	2							
-0.5	5100.	0.5	-5100.					
	3							(21)
-8.5	26900.	-0.125	80.	0.0	0.0			
	3							(22)
0.0	0.0	0.125	32.5	10.0	12700.			
19.1	0.6	386.						(23)
0.	0.	1000.						(24)
1.2	0.0005							(25)
	100							(26)
	2							(27)
	1	1						(28)
	4							(29)
30.5	16.25	8.5	1657.5			K=1	I=1, N=1	(30)
25.5	48.75	76.5	16.25					(31)
30.5	41.25	8.5	892.5			K=2		(30)
25.5	8.75	76.5	41.25					(31)
30.5	16.25	-42.5	1657.5			K=3		(30)
76.5	48.75	25.5	16.25					(31)
30.5	41.25	-42.5	892.5			K=4		(30)
76.5	8.75	25.5	41.25					(31)
	2							(29)
-15.25	25.	34.0	1525.			K=1	I=3, N=1	(30)
15.25	25.	45.75	25.					(31)
15.25	25.	34.0	1525.			K=2		(30)
45.75	25.	15.25	25.					(31)
	0						I=4, N=1	(29)
	1							(29)
0.	50.	-17.	6222.			K=1	I=5, N=1	(30)
0.	0.	0.	0.					(31)
	1							(29)
0.	0.	-17.	6222.			K=2	I=6, N=1	(30)
0.	0.	0.	0.					(31)

						Group No.
	9					(29)
44.	19.42	-103.5	1902.8	K=1		(30)
137.5	51.74	122.5	19.4			(31)
44.	58.25	-103.5	1902.8	K=2		(30)
137.5	25.4	122.5	58.25			(31)
44.	95.08	-103.5	1804.8	K=3		(30)
24.5	18.42	122.5	134.9			(31)
44.	19.42	-152.5	1902.8	K=4		(30)
186.5	86.4	73.5	19.4			(31)
44.	58.25	-152.5	1902.8	K=5		(30)
186.5	58.25	73.5	58.25		I=1, N=2	(31)
44.	95.08	-152.5	1804.8	K=6		(30)
73.5	18.42	73.5	134.9			(31)
44.	19.42	-201.5	1902.8	K=7		(30)
235.5	97.1	24.5	19.4			(31)
44.	58.25	-201.5	1902.8	K=8		(30)
235.5	58.25	24.5	58.25			(31)
44.	95.08	-201.5	1804.8	K=9		(30)
122.5	18.42	24.5	134.9			(31)
	4					(29)
-22.	90.	-79.	2178.	K=1		(30)
22.	24.75	66.	90.			(31)
22.	90.	-79.	2178.	K=2		(30)
66.	24.75	22.	90.		I=3, N=2	(31)
-37.25	25.	-79.	675.	K=3		(30)
6.75	74.5	81.25	25.			(31)
37.25	25.	-79.	675.	K=4		(30)
81.25	74.5	6.75	25.			(31)
	4					(29)
+22.	29.	-226.	2563.	K=1		(30)
22.	85.4	66.	29.1			(31)
-22.	29.	-226.	2563.	K=2		(30)
66.	85.4	22.	29.1		I=4, N=2	(31)
+22.	86.37	-226.	2475.	K=3		(30)
22.	28.1	66.	86.4			(31)
-22.	86.37	-226.	2475.	K=4		(30)
66.	28.1	22.	86.4			(31)
	1					(29)
0.	114.5	-152.5	12936.	K=1	I=5, N=2	(30)
0.	0.	0.	0.			(31)
	1					(29)
0.	0.	-152.5	12936.	K=1	I=6, N=2	(30)
0.	0.	0.	0.			(31)
	0	1				(32)
6.0	0.	0.	0.	0.8	4250.	(33)
0.						(33)
END						(1)

TABLE 14

SAMPLE PROBLEM PRELIMINARY OUTPUT

```
                         GENERALIZED TRUCK SHELTER, VERSION 2.0
                                        JOB    1


 RUN ID = M35A2 WITH RETROFIT SHELTER RIGIDLY ATTACHED, RACKS SPRING-MOUNTED.


VEHICLE TYPE - WHEELED.
SHELTER RIGIDLY MOUNTED.

NUMBER OF RACKS =  2
RACK NUMBER  1 HAS 11 SPRINGS.
RACK NUMBER  2 HAS 11 SPRINGS.

TRUCK HAS 3 AXLES.

AXLE   TIRES  SPRINGS
   1     1       2
   2     2       1
   3     2       1

INDEX             MASS              X                Y                Z

    1        .2814000E+02   0.              .3169000E+02   -.8577000E+02
    2        .3660000E+01   0.             0.               0.
    3        .4230000E+01   0.             0.              -.1300000E+03
    4        .4230000E+01   0.             0.              -.1780000E+03
    5        .1200000E+01    .2901000E+02   .7451000E+02   -.1269600E+03
    6        .1200000E+01    .2901000E+02   .7451000E+02   -.1730400E+03

SYSTEM CG MASS =    .4506000E+02 X =    0.            Y =    .2772758E+02 Z =   -.9845557E+02

INDEX =  1 VECTOR FROM CG TO MASS =        0.                    .396242E+01        .126856E+02
INDEX =  2 VECTOR FROM CG TO MASS =        0.                   -.277276E+02        .984556E+02
INDEX =  3 VECTOR FROM CG TO MASS =        0.                   -.277276E+02       -.315444E+02
INDEX =  4 VECTOR FROM CG TO MASS =        0.                   -.277276E+02       -.795444E+02
INDEX =  5 VECTOR FROM CG TO MASS =        .290100E+02           .467824E+02       -.285044E+02
INDEX =  6 VECTOR FROM CG TO MASS =        .290100E+02           .467824E+02       -.745844E+02

INERTIA DATA

  XIYZ =  -.19600E+05

INERTIA  XI1          XI2          XI3
     .11240E+06    .97400E+05   .40900E+05
    0.             .25300E+04   .25300E+04
    0.             .77300E+04   .77300E+04
    0.             .77300E+04   .77300E+04
     .54700E+03    .66600E+02   .40500E+03
     .54700E+03    .66600E+02   .40500E+03

SPRING ATTACHMENT POINTS
          1      .15400E+02     .10800E+02    0.              .15400E+02     .56000E+01    0.
          2      .15400E+02    -.14000E+01   -.32400E+01      .15400E+02     .11190E+02   -.10760E+02
          3      .20240E+02    0.            -.13000E+03      .20240E+02     .50000E+01   -.13000E+03
          4      .20240E+02    0.            -.17800E+03      .20240E+02     .50000E+01   -.17800E+03

TIRE             .33874E+02    0.            0.
TIRE-(CG)        .33874E+02    -.27728E+02    .98456E+02

TIRE             .29250E+02    0.            -.13000E+03
TIRE-(CG)        .29250E+02    -.27728E+02   -.31544E+02
```

TIRE	.40750E+02	0.	-.13000E+03
TIRE-(CG)	.40750E+02	-.27728E+02	-.31544E+02
TIRE	.29250E+02	0.	-.17800E+03
TIRE-(CG)	.29250E+02	-.27728E+02	-.79544E+02
TIRE	.40750E+02	0.	-.17800E+03
TIRE-(CG)	.40750E+02	-.27728E+02	-.79544E+02

BOGIE SPRING ATTACHMENT POINTS

SPRING	AXLE	X
5	2	.202400E+02

RACK SPRING ATTACHMENT POINTS

SPRING						
6	.29010E+02	.41870E+02	-.12073E+03	.35000E+02	.41870E+02	-.12073E+03
7	.29010E+02	.41870E+02	-.13319E+03	.35000E+02	.41870E+02	-.13319E+03
8	.29010E+02	.10237E+03	-.12135E+03	.35000E+02	.10237E+03	-.12135E+03
9	.29010E+02	.10237E+03	-.13257E+03	.35000E+02	.10237E+03	-.13257E+03
10	.34470E+02	.41870E+02	-.12093E+03	.34470E+02	.35000E+02	-.12093E+03
11	.23550E+02	.41870E+02	-.12093E+03	.23550E+02	.35000E+02	-.12093E+03
12	.34470E+02	.41870E+02	-.13299E+03	.34470E+02	.35000E+02	-.13299E+03
13	.23550E+02	.41870E+02	-.13299E+03	.23550E+02	.35000E+02	-.13299E+03
14	.22320E+02	.41870E+02	-.12696E+03	.22320E+02	.41870E+02	-.12000E+03
15	.35700E+02	.41870E+02	-.12696E+03	.35700E+02	.41870E+02	-.12000E+03
16	.37510E+02	.11087E+03	-.12696E+03	.37510E+02	.11087E+03	-.12000E+03
17	.29010E+02	.41870E+02	-.16681E+03	.35000E+02	.41870E+02	-.16681E+03
18	.29010E+02	.41870E+02	-.17927E+03	.35000E+02	.41870E+02	-.17927E+03
19	.29010E+02	.10237E+03	-.16743E+03	.35000E+02	.10237E+03	-.16743E+03
20	.29010E+02	.10237E+03	-.17865E+03	.35000E+02	.10237E+03	-.17865E+03
21	.34470E+02	.41870E+02	-.14701E+03	.34470E+02	.35000E+02	-.14701E+03
22	.23550E+02	.41870E+02	-.14701E+03	.23550E+02	.35000E+02	-.14701E+03
23	.34470E+02	.41870E+02	-.17907E+03	.34470E+02	.35000E+02	-.17907E+03
24	.23550E+02	.41870E+02	-.17907E+03	.23550E+02	.35000E+02	-.17907E+03
25	.22320E+02	.41870E+02	-.17304E+03	.22320E+02	.41870E+02	-.16000E+03
26	.35700E+02	.41870E+02	-.17304E+03	.35700E+02	.41870E+02	-.16000E+03
27	.37510E+02	.11087E+03	-.17304E+03	.37510E+02	.11087E+03	-.16000E+03

DATA FOR NON-LINEAR DAMPING

		V	F
SPRING 1	1	-.100000E+05	.654000E+04
	2	-.500000E+01	.440000E+02
	3	0.	0.
	4	.500000E+01	-.810000E+02
	5	.100000E+05	-.183200E+05
SPRING 2	6	-.100000E+05	.261600E+05
	7	-.500000E+01	.175000E+03
	8	0.	0.
	9	.500000E+01	-.325000E+03
	10	.100000E+05	-.733000E+05
SPRING 3	11	-.100000E+03	0.
	12	.100000E+03	0.
SPRING 4	13	-.100000E+03	0.

```
               14     .100000E+03   0.

SPRING  5                V                F
               15    -.100000E+05    .130600E+05
               16    -.500000E+01    .880000E+02
               17    0.              0.
               18     .500000E+01   -.162000E+03
               19     .100000E+05   -.366000E+05

SPRING  6                V                F
               20    -.100000E+05    .101400E+06
               21     .100000E+05   -.101400E+06

SPRING  7                V                F
               22    -.100000E+05    .101400E+06
               23     .100000E+05   -.101400E+06

SPRING  8                V                F
               24    -.100000E+05    .507000E+05
               25     .100000E+05   -.507000E+05

SPRING  9                V                F
               26    -.100000E+05    .507000E+05
               27     .100000E+05   -.507000E+05

SPRING 10                V                F
               28    -.100000E+05    .760500E+05
               29     .100000E+05   -.760500E+05

SPRING 11                V                F
               30    -.100000E+05    .760500E+05
               31     .100000E+05   -.760500E+05

SPRING 12                V                F
               32    -.100000E+05    .760500E+05
               33     .100000E+05   -.760500E+05

SPRING 13                V                F
               34    -.100000E+05    .760500E+05
               35     .100000E+05   -.760500E+05

SPRING 14                V                F
               36    -.100000E+05    .101400E+06
               37     .100000E+05   -.101400E+06

SPRING 15                V                F
               38    -.100000E+05    .101400E+06
               39     .100000E+05   -.101400E+06

SPRING 16                V                F
               40    -.100000E+05    .101400E+06
               41     .100000E+05   -.101400E+06

SPRING 17                V                F
               42    -.100000E+05    .101400E+06
               43     .100000E+05   -.101400E+06

SPRING 18                V                F
               44    -.100000E+05    .101400E+06
               45     .100000E+05   -.101400E+06

SPRING 19                V                F
               46    -.100000E+05    .507000E+05
```

```
              47    .100000E+05  -.507000E+05

SPRING 20              V              F
              48   -.100000E+05   .507000E+05
              49    .100000E+05  -.507000E+05

SPRING 21              V              F
              50   -.100000E+05   .760500E+05
              51    .100000E+05  -.760500E+05

SPRING 22              V              F
              52   -.100000E+05   .760500E+05
              53    .100000E+05  -.760500E+05

SPRING 23              V              F
              54   -.100000E+05   .760500E+05
              55    .100000E+05  -.760500E+05

SPRING 24              V              F
              56   -.100000E+05   .760500E+05
              57    .100000E+05  -.760500E+05

SPRING 25              V              F
              58   -.100000E+05   .101400E+06
              59    .100000E+05  -.101400E+06

SPRING 26              V              F
              60   -.100000E+05   .101400E+06
              61    .100000E+05  -.101400E+06

SPRING 27              V              F
              62   -.100000E+05   .101400E+06
              63    .100000E+05  -.101400E+06

TIRE DAMPING COEFFICIENTS
  NORMAL          .500000E+01
  TANGENTIAL      .500000E+01

                                DATA FOR NON-LINEAR SPRINGS.

SPRING  1              X              F(X)
               1   -.62000E+01    .62000E+04
               2   0.            0.
               3    .10000E+03   -.10000E+06

SPRING  2              X              F(X)
               1   -.10000E+03   0.
               2    .10000E+03   0.

SPRING  3              X              F(X)
               1   -.50000E+01   0.
               2    .10000E+03   0.

SPRING  4              X              F(X)
               1   -.50000E+01   0.
               2    .10000E+03   0.

SPRING  5              X              F(X)
               1   -.62000E+01    .71300E+04
               2   0.            0.
               3    .10000E+03   -.11500E+06

SPRING  6              X              F(X)
```

```
                 1     -.50000E+00     .51000E+04
                 2      .50000E+00    -.51000E+04

SPRING  7               X               F(X)
                 1     -.50000E+00     .51000E+04
                 2      .50000E+00    -.51000E+04

SPRING  8               X               F(X)
                 1     -.50000E+00     .25500E+04
                 2      .50000E+00    -.25500E+04

SPRING  9               X               F(X)
                 1     -.50000E+00     .25500E+04
                 2      .50000E+00    -.25500E+04

SPRING 10               X               F(X)
                 1     -.50000E+00     .38250E+04
                 2      .50000E+00    -.38250E+04

SPRING 11               X               F(X)
                 1     -.50000E+00     .38250E+04
                 2      .50000E+00    -.38250E+04

SPRING 12               X               F(X)
                 1     -.50000E+00     .38250E+04
                 2      .50000E+00    -.38250E+04

SPRING 13               X               F(X)
                 1     -.50000E+00     .38250E+04
                 2      .50000E+00    -.38250E+04

SPRING 14               X               F(X)
                 1     -.50000E+00     .51000E+04
                 2      .50000E+00    -.51000E+04

SPRING 15               X               F(X)
                 1     -.50000E+00     .51000E+04
                 2      .50000E+00    -.51000E+04

SPRING 16               X               F(X)
                 1     -.50000E+00     .51000E+04
                 2      .50000E+00    -.51000E+04

SPRING 17               X               F(X)
                 1     -.50000E+00     .51000E+04
                 2      .50000E+00    -.51000E+04

SPRING 18               X               F(X)
                 1     -.50000E+00     .51000E+04
                 2      .50000E+00    -.51000E+04

SPRING 19               X               F(X)
                 1     -.50000E+00     .25500E+04
                 2      .50000E+00    -.25500E+04

SPRING 20               X               F(X)
                 1     -.50000E+00     .25500E+04
                 2      .50000E+00    -.25500E+04

SPRING 21               X               F(X)
                 1     -.50000E+00     .38250E+04
                 2      .50000E+00    -.38250E+04
```

SPRING 22

	X	F(X)
1	-.50000E+00	.38250E+04
2	.50000E+00	-.38250E+04

SPRING 23

	X	F(X)
1	-.50000E+00	.38250E+04
2	.50000E+00	-.38250E+04

SPRING 24

	X	F(X)
1	-.50000E+00	.38250E+04
2	.50000E+00	-.38250E+04

SPRING 25

	X	F(X)
1	-.50000E+00	.51000E+04
2	.50000E+00	-.51000E+04

SPRING 26

	X	F(X)
1	-.50000E+00	.51000E+04
2	.50000E+00	-.51000E+04

SPRING 27

	X	F(X)
1	-.50000E+00	.51000E+04
2	.50000E+00	-.51000E+04

SPRING 28

	X	F(X)
1	-.85000E+01	.26900E+05
2	-.12500E+00	.80000E+02
3	0.	0.

SPRING 29

	X	F(X)
1	0.	0.
2	.12500E+00	.32500E+02
3	.10000E+02	.12700E+05

RADIUS OF WHEEL = .19100E+02 COEFFICIENT OF SLIDING FRICTION = .80000E+00 GRAVITY = .38600E+03
GROUND SLOPES ARE GS(1) = 0. GS(2) = 0. THE POSITION OF THE SLOPE CHANGE IS .10000E+04

PROGRAM WILL RUN UNTIL .12000E+01 SECONDS OF RESPONSE HAVE BEEN COMPUTED.
THE TIME INTERVAL BETWEEN STEPS = .50000E-03 SECONDS.
PRINT TIME HISTORY RESPONSE EVERY 100 STEPS.

A SPECIAL TIME INTERVAL OF .15000E-03 SECONDS IS USED UNTIL .30000E-01 SECONDS.

NUMBER OF AERODYNAMIC BOX CONFIGURATIONS = 2

BOX DESIGNATION CODE

BOX NO.	CODE	
1	1	BOX RIGIDLY ATTACHED TO VEHICLE
2	1	BOX RIGIDLY ATTACHED TO VEHICLE

BOX	FACE	GRID POINT	X	Y	Z	AREA	SI1	SI2	SI3	SI4
1	1	1	.305000E+02	.162500E+02	.850000E+01	.165750E+04	.255000E+02	.487500E+02	.765000E+02	.162500E+02
1	1	2	.305000E+02	.412500E+02	.850000E+01	.892500E+03	.255000E+02	.875000E+01	.765000E+02	.412500E+02
1	1	3	.305000E+02	.162500E+02	-.425000E+02	.165750E+04	.765000E+02	.487500E+02	.255000E+02	.162500E+02
1	1	4	.305000E+02	.412500E+02	-.425000E+02	.892500E+03	.765000E+02	.875000E+01	.255000E+02	.412500E+02
1	3	1	-.152500E+02	.250000E+02	.340000E+02	.152500E+04	.152500E+02	.250000E+02	.457500E+02	.250000E+02
1	3	2	.152500E+02	.250000E+02	.340000E+02	.152500E+04	.457500E+02	.250000E+02	.152500E+02	.250000E+02
1	5	1	0.	.500000E+02	-.170000E+02	.622200E+04	0.	0.	0.	0.
1	6	1	0.	0.	-.170000E+02	.622200E+04	0.	0.	0.	0.
2	1	1	.440000E+02	.194200E+02	-.1[illegible]500E+03	.190280E+04	.137500E+03	.517400E+02	.122500E+03	.194000E+02
2	1	2	.440000E+02	.582500E+02	-.1[illegible]500E+03	.190280E+04	.137500E+03	.254000E+02	.122500E+03	.582500E+02
2	1	3	.440000E+02	.950800E+02	-.1[illegible]500E+03	.180480E+04	.245000E+02	.184200E+02	.122500E+03	.154400E+03
2	1	4	.440000E+02	.194200E+02	-.1[illegible]500E+03	.190280E+04	.186500E+03	.864000E+02	.735000E+02	.194000E+02

2	1	5	.440000E+02	.582500E+02	-.152500E+03	.190280E+04	.186500E+03	.582500E+02	.735000E+02	.582500E+02
2	1	6	.440000E+02	.950800E+02	-.152500E+03	.180480E+04	.735000E+02	.184200E+02	.735000E+02	.134900E+03
2	1	7	.440000E+02	.194200E+02	-.201500E+03	.190280E+04	.235500E+03	.971000E+02	.245000E+02	.194000E+02
2	1	8	.440000E+02	.582500E+02	-.201500E+03	.190280E+04	.235500E+03	.582500E+02	.245000E+02	.582500E+02
2	1	9	.440000E+02	.950800E+02	-.201500E+03	.180480E+04	.122500E+03	.184200E+02	.245000E+02	.134900E+03
2	3	1	-.220000E+02	.900000E+02	-.790000E+02	.217800E+04	.220000E+02	.247500E+02	.660000E+02	.900000E+02
2	3	2	.220000E+02	.900000E+02	-.790000E+02	.217800E+04	.660000E+02	.247500E+02	.220000E+02	.900000E+02
2	3	3	-.372500E+02	.250000E+02	-.790000E+02	.675000E+03	.675000E+01	.745000E+02	.812500E+02	.250000E+02
2	3	4	.372500E+02	.250000E+02	-.790000E+02	.675000E+03	.812500E+02	.745000E+02	.675000E+01	.250000E+02
2	4	1	.220000E+02	.290000E+02	-.224000E+03	.256300E+04	.220000E+02	.854000E+02	.660000E+02	.291000E+02
2	4	2	-.220000E+02	.290000E+02	-.224000E+03	.256300E+04	.660000E+02	.854000E+02	.220000E+02	.291000E+02
2	4	3	.220000E+02	.863700E+02	-.224000E+03	.247500E+04	.220000E+02	.281000E+02	.660000E+02	.864000E+02
2	4	4	-.220000E+02	.863700E+02	-.224000E+03	.247500E+04	.660000E+02	.281000E+02	.220000E+02	.864000E+02
2	5	1	0.	.114500E+03	-.152500E+03	.129360E+05	0.	0.	0.	0.
2	6	1	0.	0.	-.152500E+03	.129360E+05	0.	0.	0.	0.

START TRIM CALCULATIONS.
VEHICLE HAS TRIMMED.

BLAST MODEL - 1 KT, SL

RESPONSE RUN ONLY

BLAST DATA

```
  ESTIMATED PSO, PSI         =    .600000E+01
  AZIMUTHAL ANGLE 1, DEG     =   0.
  AZIMUTHAL ANGLE 2, DEG     =   0.
  INCREMENT IN ANGLE, DEG    =   0.
  YIELD, KT                  =    .600000E+00
  ALTITUDE, FT               =    .425000E+04
```

AMBIENT CONDITIONS

```
  PRESSURE, PSI          =    .125753E+02
  SPEED OF SOUND, FT/SEC =    .110003E+04
```

TABLE 15

SAMPLE PROBLEM TIME-HISTORY OUTPUT EXCERPTS

```
JOB  1    CASE  1    TRIAL  1
  AZIMUTHAL ANGLE, DEG                =     0.
  PEAK OVERPRESSURE, PSI              =     .600000E+01
  PEAK DYNAMIC PRESSURE, PSI          =     .957171E+00
  RANGE, METERS                       =     .354353E+03
  OVERPRESSURE IMPULSE, PSI-SEC       =     .579712E+00
  DYNAMIC PRESSURE IMPULSE, PSI-SEC   =     .609171E-01

BEGIN TIME HISTORY CALCULATIONS.


TIME =    0.                  CG DISTANCE =     .376542E+02

DISPLACEMENT =
  0.          0.          -.3792E+01  -.2967E-02  0.          -.7798E-01  0.           .2422E+01  0.           .3351E+01
  0.           .3493E+01  -.2293E-16  -.2763E-14  -.1514E-01  -.7147E-06  -.2679E-05  -.5939E-04  -.1678E-16  -.3069E-14
 -.1651E-01  -.1371E-03  -.2188E-03  -.1987E-02   .2287E-16   .2768E-14  -.1514E-01  -.7147E-06   .2679E-05  -.5939E-04
 -.7479E-18   .4205E-14  -.1651E-01  -.1371E-03   .2188E-03  -.1987E-02

VELOCITY =
  0.          0.          0.          0.          0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.

ACCELERATION =
   .1507E+03  -.7230E+04  -.2335E-11  -.9506E-13   .6713E+02   .2541E-11  -.1507E+03  -.1226E-10  -.1507E+03  -.7705E-04
 -.1507E+03   .7705E-04  -.1507E+03   .1619E+05  -.4371E+04   .1892E-11  -.6713E+02   .1948E+04  -.1507E+03   .1929E+05
 -.4371E+04   .9119E-12  -.6713E+02   .1948E+04  -.1507E+03   .1619E+05   .4371E+04   .1874E-11  -.6713E+02  -.1948E+04
 -.1507E+03   .1929E+05   .4371E+04  -.1405E-11  -.6713E+02  -.1948E+04

ROLL =  0.           , PITCH =  -.29669E-02, YAW =  0.           ,   X =  0.          , Y =  -.37917E+01, Z =  -.72357E-01
AT TIME =    .780000E-02  SPRING 27  BOTTOMS OUT, DEFLECTION =  -.507211E+00
AT TIME =    .780000E-02  SPRING 48  BOTTOMS OUT, DEFLECTION =   .501807E+00
AT TIME =    .780000E-02  SPRING 49  BOTTOMS OUT, DEFLECTION =   .504750E+00
AT TIME =    .795000E-02  SPRING 49  BOTTOMS OUT, DEFLECTION =   .510169E+00
AT TIME =    .855000E-02  SPRING 15  BOTTOMS OUT, DEFLECTION =  -.502803E+00
AT TIME =    .855000E-02  SPRING 16  BOTTOMS OUT, DEFLECTION =  -.507460E+00
AT TIME =    .855000E-02  SPRING 37  BOTTOMS OUT, DEFLECTION =   .503550E+00
AT TIME =    .855000E-02  SPRING 38  BOTTOMS OUT, DEFLECTION =   .507354E+00
AT TIME =    .870000E-02  SPRING 16  BOTTOMS OUT, DEFLECTION =  -.511222E+00
AT TIME =    .870000E-02  SPRING 38  BOTTOMS OUT, DEFLECTION =   .511432E+00
AT TIME =    .108000E-01  SPRING 24  BOTTOMS OUT, DEFLECTION =  -.500415E+00
AT TIME =    .115500E-01  SPRING 47  BOTTOMS OUT, DEFLECTION =   .501158E+00

TIME =     .5000E-01          CG DISTANCE =     .360368E+02

DISPLACEMENT =
   .3726E-01  -.2343E+01  -.3737E+01  -.1459E-02   .9366E-02   .7365E-01  -.3236E-01   .2491E+01  -.3116E-01   .3280E+01
 -.3036E-01   .3357E+01  -.2737E-02   .1096E+00  -.2857E-01   .4037E-03   .2246E-02   .6241E-01  -.1956E-02   .1454E+00
 -.4789E-01  -.7884E-03  -.1664E-02  -.2399E-01  -.2690E-02   .1078E+00   .9470E-02  -.1572E-03   .1694E-02  -.3906E-01
 -.2406E-02   .1193E+00  -.8816E-02  -.5790E-03   .1992E-02  -.4055E-01

VELOCITY =
   .9042E+00  -.5123E+02   .2391E+01   .4797E-01   .1410E+00   .4974E+01  -.7275E+00   .1373E+01  -.6502E+00  -.2574E+01
 -.6480E+00  -.4509E+01   .5311E+00  -.1123E+02   .8774E+01   .1232E+00   .7224E+00  -.4016E+01   .9296E+00  -.2658E+02
   .1016E+02   .7945E-01  -.2548E+00   .8013E+01   .5047E+00  -.1044E+02  -.6758E+01  -.3273E-01   .2187E+00   .7428E+01
   .8438E+00  -.1967E+02  -.2534E+01  -.2499E-01   .2740E+00   .1221E+02

ACCELERATION =
 -.3452E+01   .3227E+03   .9597E+02   .1641E+01  -.5272E+01   .9364E+02   .4487E+01   .6275E+02   .3622E+01  -.1268E+03
   .2134E+01  -.1998E+03   .7744E+02  -.2413E+04   .1861E+03   .1204E+02  -.1721E+02  -.1489E+04  -.3077E+02  -.3555E+04
   .4216E+03   .1700E+02   .2949E+02  -.2530E+03   .7677E+02  -.2476E+04  -.5904E+03  -.1725E+02  -.2384E+02   .5898E+03
```

```
TIME =     .8000E+00          CG DISTANCE =     .193007E+02

DISPLACEMENT =
   .4023E+00  -.1649E+02   .7160E+01   .4999E-01   .3188E-01   .9956E+00  -.2605E+00   .1863E+01  -.1895E+00   .4464E+00
  -.9312E-01   .1777E+01   .1375E-03  -.9546E-02  -.3807E-02   .5586E-04   .1364E-03   .3070E-02   .1313E-03  -.9051E-02
  -.7375E-03   .3904E-04   .1232E-03   .2976E-02   .1386E-03  -.9629E-02  -.6922E-02   .5549E-04  -.1459E-03   .2698E-02
   .1328E-03  -.9162E-02  -.4143E-02   .6349E-05  -.6629E-04   .1932E-02

VELOCITY =
   .1725E+00   .1627E+01  -.1187E+02  -.1643E+00   .1018E-01   .4538E+01  -.9268E-01  -.1277E+01  -.5562E+00   .1234E+01
  -.3329E+00   .8345E+01   .1295E-03  -.1260E-01  -.4518E-01  -.5870E-04   .9237E-03  -.5551E-03   .5128E-03  -.4144E-01
  -.7414E-01  -.6206E-03   .3334E-03  -.2875E-02   .8383E-04  -.9546E-02  -.3855E-01   .1723E-03  -.1499E-02   .2232E-01
   .4389E-03  -.3674E-01  -.7989E-01  -.9587E-03   .1752E-02  -.1280E-01

ACCELERATION =
  -.1370E+01  -.1172E+02  -.1731E+03  -.1421E+01   .2313E+00   .2488E+02   .2004E+01  -.8140E+01   .2829E+01   .2134E+03
  -.3060E+01   .4002E+03  -.1361E-01   .7763E+00  -.1462E+01   .1603E+00   .7694E+00   .7346E+01  -.7948E-02   .1109E+00
  -.3631E+01   .4028E-01   .3939E+00   .3273E+01  -.1362E-01   .7722E+00  -.1758E+01   .3927E-01  -.2313E+00   .1894E+01
  -.1683E-01   .6296E+00  -.4632E+01   .1007E-01  -.2466E+00   .2698E+01

ROLL =   .40317E+00, PITCH =   .54683E-01, YAW =   .24807E-01,   X =  -.19446E+02, Y =   .32783E+01, Z =   .15907E+01

TIME =     .8500E+00          CG DISTANCE =     .193534E+02

DISPLACEMENT =
   .4092E+00  -.1643E+02   .6367E+01   .4043E-01   .3245E-01   .1238E+01  -.2631E+00   .1814E+01  -.2136E+00   .6934E+00
  -.1147E+00   .2651E+01   .1471E-03  -.1034E-01  -.6831E-02   .2112E-04   .6389E-04   .1309E-02   .1555E-03  -.1104E-01
  -.6387E-02  -.3443E-04  -.1344E-04   .6075E-03   .1474E-03  -.1036E-01  -.1006E-01   .3426E-04  -.1028E-03   .2368E-02
   .1559E-03  -.1107E-01  -.9788E-02  -.5596E-04   .3998E-04   .1039E-02

VELOCITY =
   .1040E+00   .9796E+00  -.1941E+02  -.2095E+00   .7604E-02   .4846E+01  -.3948E-01  -.1561E+00  -.4110E+00   .7211E+01
  -.5263E+00   .2443E+02   .1990E-03  -.2062E-01  -.8378E-01  -.4501E-03  -.1085E-03  -.2297E-01   .4952E-03  -.4779E-01
  -.1269E+00  -.1049E-02   .2369E-03  -.1168E-01   .1739E-03  -.1892E-01  -.7558E-01  -.1670E-03  -.2153E-03   .1497E-01
   .4308E-03  -.4379E-01  -.1302E+00  -.1311E-02   .2072E-02  -.6295E-02

ACCELERATION =
  -.1392E+01  -.1292E+02  -.1226E+03  -.4035E+00  -.3459E+00  -.1230E+02  -.8407E+00   .4592E+02   .2410E+01   .4418E+02
  -.2517E+01   .1358E+03   .3177E-01  -.9980E+00  -.1506E+01   .9093E-01   .3989E+00   .5480E+01   .6208E-01  -.2173E+01
  -.2139E+01  -.3140E-02   .3711E-01   .1664E+01   .3361E-01  -.1111E+01  -.3436E+00   .1719E-01  -.4992E-01   .2451E+01
   .6212E-01  -.2181E+01  -.1728E+01   .2968E-04   .1485E-01   .1889E+01

ROLL =   .41012E+00, PITCH =   .45303E-01, YAW =   .25018E-01,   X =  -.19062E+02, Y =   .25683E+01, Z =   .17915E+01
AT TIME =   .898500E+00  SPRING 10  BOTTOMS OUT, DEFLECTION =  -.620237E+01
```

TABLE 16

SAMPLE PROBLEM SUMMARY INFORMATION

SUMMARY INFORMATION

DISPLACEMENTS

DOF	.....MAXIMUM	.......TIME	......MINIMUM	........TIME
1	.41263E+00	.89850E+00	0.	0.
2	0.	0.	-.16683E+02	.67750E+00
3	.75654E+01	.73050E+00	-.37916E+01	.19500E-02
4	.57679E-01	.70650E+00	-.29670E-02	.19500E-02
5	.39212E-01	.56600E+00	0.	0.
6	.19797E+01	.35200E+00	-.78081E-01	.19500E-02
7	0.	0.	-.32113E+00	.34900E+00
8	.28496E+01	.18300E+00	.11414E+01	.33550E+00
9	0.	0.	-.27776E+00	.10135E+01
10	.39528E+01	.12000E+01	.44295E+00	.79450E+00
11	0.	0.	-.26460E+00	.23850E+00
12	.37880E+01	.89850E+00	.10938E+00	.11430E+01
13	.37526E-02	.33000E-01	-.46944E-02	.85500E-02
14	.37863E+00	.93000E-02	-.18525E+00	.27300E-01
15	.85752E-01	.23850E+00	-.11513E+00	.10650E-01
16	.11494E-02	.25750E+00	-.20338E-02	.22800E-01
17	.53238E-02	.57000E-01	-.13418E-01	.15750E-01
18	.67342E-01	.47500E-01	-.97214E-01	.27900E-01
19	.63397E-02	.29700E-01	-.63840E-02	.43000E-01
20	.38320E+00	.85500E-02	-.25989E+00	.24600E-01
21	.99349E-01	.23850E+00	-.11987E+00	.10800E-01
22	.10159E-02	.26250E+00	-.17458E-02	.27700E+00
23	.12648E-01	.13200E-01	-.71312E-02	.29550E-01
24	.63681E-01	.26050E+00	-.77839E-01	.40500E-01
25	.38200E-02	.33000E-01	-.46958E-02	.85500E-02
26	.37989E+00	.94500E-02	-.18499E+00	.27600E-01
27	.79059E-01	.10350E-01	-.14040E+00	.30200E+00
28	.14476E-02	.22350E-01	-.11398E-02	.36500E-01
29	.43968E-02	.31000E-01	-.11946E-01	.15300E-01
30	.51554E-01	.26700E-01	-.64130E-01	.44000E-01
31	.51305E-02	.31500E-01	-.53132E-02	.44500E-01
32	.38960E+00	.87000E-02	-.22569E+00	.25800E-01
33	.87577E-01	.29100E+00	-.15670E+00	.30200E+00
34	.19379E-02	.18300E-01	-.21949E-02	.27650E+00
35	.68499E-02	.10030E+01	-.14821E-01	.14100E-01
36	.70882E-01	.25200E-01	-.95881E-01	.42500E-01

VELOCITIES

DOF	.....MAXIMUM	.......TIME	......MINIMUM	........TIME
1	.14569E+01	.17800E+00	-.78020E+00	.11990E+01
2	.27059E+02	.12000E+01	-.56070E+02	.35000E-01
3	.41590E+02	.31800E+00	-.23740E+02	.89800E+00
4	.27976E+00	.31750E+00	-.21323E+00	.86900E+00
5	.23124E+00	.30500E-01	-.15205E+00	.10210E+01
6	.87936E+01	.21700E+00	-.55498E+01	.56900E+00
7	.71954E+00	.42150E+00	-.14266E+01	.20550E+00
8	.16284E+02	.38900E+00	-.34023E+02	.20600E+00
9	.10141E+01	.46150E+00	-.15205E+01	.14200E+00
10	.51669E+02	.26200E+00	-.59722E+02	.23850E+00
11	.12818E+01	.43700E+00	-.15211E+01	.14100E+00
12	.48463E+02	.23700E+00	-.48728E+02	.26250E+00
13	.17572E+01	.87000E-02	-.91641E+00	.39000E-01
14	.60619E+02	.63000E-02	-.51015E+02	.17850E-01

15	.23476E+02	.23450E+00	-.18136E+02	.29200E+00
16	.27601E+00	.29250E-01	-.36134E+00	.26250E+00
17	.16617E+01	.22050E-01	-.44625E+01	.85500E-02
18	.13335E+02	.37500E-01	-.15858E+02	.26450E+00
19	.28084E+01	.78000E-02	-.15098E+01	.36000E-01
20	.66996E+02	.63000E-02	-.63121E+02	.15150E-01
21	.31983E+02	.23450E+00	-.26808E+02	.26400E+00
22	.27832E+00	.23600E+00	-.29073E+00	.26800E+00
23	.44801E+01	.78000E-02	-.19374E+01	.20100E-01
24	.93605E+01	.23450E+00	-.16415E+02	.26450E+00
25	.16669E+01	.87000E-02	-.90326E+00	.39000E-01
26	.60679E+02	.63000E-02	-.50833E+02	.18000E-01
27.	.20668E+02	.31000E+00	-.28212E+02	.29300E+00
28	.21552E+00	.16800E-01	-.32832E+00	.26250E+00
29	.17002E+01	.21750E-01	-.42416E+01	.85500E-02
30	.10411E+02	.23450E+00	-.14151E+02	.26450E+00
31	.24324E+01	.79500E-02	-.12640E+01	.37500E-01
32	.67056E+02	.63000E-02	-.58271E+02	.16500E-01
33	.27500E+02	.23450E+00	-.34665E+02	.29250E+00
34	.33359E+00	.28350E+00	-.37224E+00	.24900E-01
35	.20372E+01	.21000E-01	-.58048E+01	.78000E-02
36	.12390E+02	.51000E-01	-.16028E+02	.26400E+00

ACCELERATIONS

DOF	.....MAXIMUM	.......TIME	......MINIMUM	.......TIME
1	.15935E+03	.90000E-03	-.13555E+02	.44000E-01
2	.43201E+03	.45500E-01	-.87817E+04	.90000E-03
3	.38294E+03	.23800E+00	-.82660E+03	.30000E+00
4	.68760E+01	.23800E+00	-.95283E+01	.27150E+00
5	.67343E+02	.45000E-03	-.60446E+01	.45500E-01
6	.33313E+03	.15000E-02	-.26203E+03	.27100E+00
7	.15480E+02	.44000E-01	-.15836E+03	.90000E-03
8	.17220E+04	.25700E+00	-.36134E+03	.95600E+00
9	.16604E+02	.43500E-01	-.15894E+03	.90000E-03
10	.33526E+04	.27150E+00	-.29553E+03	.58050E+00
11	.15177E+02	.43500E-01	-.15879E+03	.90000E-03
12	.36437E+04	.27200E+00	-.62011E+03	.55500E+00
13	.20646E+03	.22350E-01	-.35604E+03	.12150E-01
14	.16798E+05	.90000E-03	-.82602E+04	.11400E-01
15	.27743E+04	.30000E+00	-.44785E+04	.90000E-03
16	.67615E+02	.22800E-01	-.48666E+02	.12000E-01
17	.54060E+03	.13800E-01	-.25313E+03	.28050E-01
18	.20826E+04	.28800E-01	-.19100E+04	.46000E-01
19	.37983E+03	.16900E-01	-.60565E+03	.10650E-01
20	.19286E+05	0.	-.86610E+04	.10650E-01
21	.36029E+04	.27150E+00	-.44731E+04	.90000E-03
22	.73006E+02	.10650E-01	-.60526E+02	.20700E-01
23	.27516E+03	.26700E-01	-.58274E+03	.10650E-01
24	.20308E+04	.45000E-03	-.17052E+04	.26000E+00
25	.20052E+03	.22350E-01	-.33828E+03	.12000E-01
26	.16800E+05	.90000E-03	-.83141E+04	.11400E-01
27	.44844E+04	.90000E-03	-.28464E+04	.31550E+00
28	.53727E+02	.26750E+00	-.64983E+02	.22800E-01
29	.52672E+03	.13650E-01	-.24550E+03	.27750E-01
30	.17572E+04	.27100E+00	-.19475E+04	0.
31	.32119E+03	.20700E-01	-.54201E+03	.11400E-01
32	.19286E+05	0.	-.88074E+04	.10650E-01
33	.49481E+04	.30050E+00	-.34266E+04	.23800E+00
34	.77636E+02	.27650E+00	-.84860E+02	.18900E-01
35	.61567E+03	.11400E-01	-.27164E+03	.27450E-01
36	.25571E+04	.42000E-01	-.26860E+04	.25950E-01

SPRING	MAX DEFLECTION	TIME	MIN DEFLECTION	TIME
1	.37527E+01	.33950E+00	-.24224E+01	0.
2	-.24224E+01	0.	-.62092E+01	.20600E+00
3	.20797E+01	0.	-.32218E+01	.33950E+00
4	.53307E+01	.20600E+00	.20797E+01	0.
5	.33505E+01	0.	-.50101E+01	.25450E+00
6	.72166E+01	.11680E+01	.33505E+01	0.
7	.34929E+01	0.	-.50070E+01	.23450E+00
8	.66410E+01	.89850E+00	.27550E+01	.66250E+00
9	.50101E+01	.25450E+00	-.33505E+01	0.
10	-.33505E+01	0.	-.72166E+01	.11680E+01
11	.50070E+01	.23450E+00	-.34929E+01	0.
12	-.27550E+01	.66250E+00	-.66410E+01	.89850E+00
13	.20230E+00	.24000E-01	-.28424E+00	.11550E-01
14	.17580E+00	.25500E-01	-.41005E+00	.12750E-01
15	.23385E+00	.30000E-01	-.50280E+00	.85500E-02
16	.25655E+00	.30000E-01	-.51122E+00	.87000E-02
17	.86449E-01	.23850E+00	-.13619E+00	.87000E-02
18	.83485E-01	.23850E+00	-.11122E+00	.12000E-01
19	.88020E-01	.23850E+00	-.13543E+00	.87000E-02
20	.85056E-01	.23850E+00	-.11227E+00	.12300E-01
21	.68381E-01	.15750E-01	-.66683E-01	.48500E-01
22	.87720E-01	.31500E-01	-.11115E+00	.15600E-01
23	.15436E+00	.27150E-01	-.10498E+00	.13200E-01
24	.26988E+00	.21900E-01	-.50041E+00	.10800E-01
25	.28956E+00	.20550E-01	-.36033E+00	.10800E-01
26	.41578E+00	.27750E-01	-.49885E+00	.78000E-02
27	.34126E+00	.27450E-01	-.50721E+00	.78000E-02
28	.89697E-01	.23850E+00	-.11899E+00	.78000E-02
29	.84515E-01	.23850E+00	-.11818E+00	.10800E-01
30	.10485E+00	.23850E+00	-.13283E+00	.78000E-02
31	.99470E-01	.23850E+00	-.13671E+00	.10800E-01
32	.51848E-01	.38000E-01	-.12082E+00	.12300E-01
33	.54651E-01	.15000E-01	-.62918E-01	.30900E+00
34	.10910E+00	.41500E-01	-.89431E-01	.26250E+00
35	.28496E+00	.11550E-01	-.18313E+00	.24000E-01
36	.39988E+00	.12600E-01	-.18422E+00	.25500E-01
37	.50355E+00	.85500E-02	-.22678E+00	.30000E-01
38	.51143E+00	.87000E-02	-.27535E+00	.30500E-01
39	.10228E+00	.87000E-02	-.14939E+00	.30200E+00
40	.74027E-01	.12000E-01	-.13705E+00	.30200E+00
41	.10066E+00	.87000E-02	-.14375E+00	.30200E+00
42	.72837E-01	.12150E-01	-.13142E+00	.30200E+00
43	.59790E-01	.46000E-01	-.45169E-01	.15300E-01
44	.11471E+00	.15450E-01	-.75450E-01	.31500E-01
45	.12342E+00	.13500E-01	-.10290E+00	.26700E-01
46	.34398E+00	.11100E-01	-.26321E+00	.22200E-01
47	.50116E+00	.11550E-01	-.24080E+00	.23550E-01
48	.50181E+00	.78000E-02	-.27610E+00	.28500E-01
49	.51017E+00	.79500E-02	-.34005E+00	.29100E-01
50	.89399E-01	.79500E-02	-.13541E+00	.30100E+00
51	.74423E-01	.23850E+00	-.12519E+00	.30100E+00
52	.96181E-01	.29100E+00	-.16935E+00	.27350E+00
53	.92508E-01	.29100E+00	-.15962E+00	.27350E+00
54	.89851E-01	.42500E-01	-.68083E-01	.30300E+00
55	.16322E+00	.14250E-01	-.12201E+00	.29100E-01
56	.11705E+00	.27350E+00	-.98147E-01	.25050E-01

VEHICLE REMAINED UPRIGHT

INITIAL CG DISTANCE = .376542E+02

MINIMUM CG DISTANCE = .192676E+02

TIME OF MINIMUM = .819000E+00

CRIT = .511698E+00

*** NORMAL END OF JOB

REFERENCES

1. Ethridge, Noel H., Blast Overturning Model for Ground Targets, BRL Report No. 1889, June, 1976. (AD #B012102L)

2. Etkin, Bernard, "Dynamics of Flight", John Wiley and Sons, Inc., New York, 1959.

3. Ethridge, Noel H., private communication.

4. Taylor, W. J., A Method for Predicting Blast Loads During the Diffraction Phase, Shock and Vibration Bulletin 42, Part 4, pg. 135, January, 1972.

5. Lee, William N. and Mente, Lawrence J., NOVA-2--A Digital Computer Program for Analyzing Nuclear Overpressure Effects on Aircraft, Part 1, Theory, AFWL-TR75-262, Pt 1, August, 1976.

6. Hoerner, S. F., "Fluid-Dynamic Drag", 1965.

7. Hankins, Dorris M., Experimental Pressure Distributions and Force Coefficients on Block Forms for Varying Mach Number, Reynolds Number, and Yaw Angle, SC-4204(TR), January, 1959.

8. Hobbs, Norman P., Zartarian, Garabed, and Walsh, John P., A Digital Computer Program for Calculating the Blast Response of Aircraft to Nuclear Explosions, Volume 1, Program Description, AFWL-TR-70-140, Vol. 1, April, 1971.

9. Radkowski, Peter, Letter to Noel Ethridge, BRL, dated 24 June, 1976.

APPENDIX A

CROSS-REFERENCE LISTING OF ALL PROGRAM VARIABLES

********** SUPER INDEX **********

SYMBOL - ================== ROUTINES IN WHICH THE SYMBOL APPEARS ==================
(AN * FOLLOWS ALL ROUTINE NAMES IN WHICH THE SYMBOL IS USED.)

SYMBOL									
A	-	AERO	DIFFL	ENGULF*	HYTRUK	INLOAD	LAMBDA*	MATXIN*	MOTION
		MULT*	SETA*	SOLVE*	SUMTAB	TIRES*	TRUCK		
AA	-	SETA*							
AB	-	SETA*							
ABS	-	ENGULF*	INT1*	MATXIN*	PITER*	RELAX*	SETA*	SOLVE*	TIRES*
ACOS	-	AERO*							
AD	-	AERO	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB*
		TRUCK*							
ADAM	-	ADAMI*	ADAM5*	BOTTOM*	EXTRNL*	MOTION*	REBOUN*	TIRES*	
ADAMI	-	MOTION*							
ADAM5	-	MOTION*							
AD1	-	AERO	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB
		TRUCK*							
AD2	-	AERO	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB
		TRUCK*							
AERO	-	EXTRNL*	TRUCK*						
AL	-	BLDATA	AERO*	DIFFL	ENGULF	INLOAD	MOTION	SETA*	TIRES*
		TRUCK							
ALM	-	ATMOS*							
ALOG	-	ATMOS*	HYTRUK*						
ALPHA	-	DRAGL*							
ALT	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB
		TRUCK*							
AM	-	TIRES*							
AMAX	-	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK			
AMAX1	-	DIFFL*	MAXI*	PITER*					
AMIN	-	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK			
AMIN1	-	GEOM*	MAXI*	PITER*					
AMPY	-	MATXIN*							
AMU	-	ATMOS*							
AMUZ	-	ATMOS*							
AN	-	TIRES*							
ANGLE	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
ATMOS	-	AERO*							
AVSLO	-	PRETRM*							

A0	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB
		TRUCK*							
A00	-	TRUCK*							
A1	-	PITER*	SETA*						
A2	-	PITER*	SETA*						
A6	-	BLDATA	AERO	DIFFL	ENGULF	INLOAD	MOTION	SETA*	TRUCK
B	-	LAMBDA*	MULT*	SOLVE*					
BASE	-	PRETRM*							
BATA	-	TIRES*							
BATAT	-	TIRES*							
BETA	-	AERO*	BASE	BETAIJ*	BOTTOM	EULER	EXTRNL*	FORCEI	FSPRNG
		GEOM	GRAV*	MASSM	MAXI	MOTION	ORIGIN*	PRETRM	REBOUN
		RFORCE	SFORCE	SUMARY	SUMTAB	TIRES*	TRIM*	TRUCK	
BETAIJ	-	MOTION*	TRIM*	TRUCK*					
BLAST	-	AERO*	DIFFL*	ENGULF*	HYTRUK*	INLOAD*	MOTION*	SETA*	SUMTAB*
		TRUCK*							
BOTM	-	BOTTOM*	INPSPR*	TRIM*	TRUCK*				
BOTTOM	-	MOTION*							
B1	-	PITER*							
B2	-	PITER*							
C	-	INT1*	LAMBDA*	PITER*					
CA	-	ATMOS*							
CALC	-	AERO*	BASE*	BETAIJ*	BOTTOM*	EULER*	EXTRNL*	FORCEI*	FSPRNG*
		GEOM*	MASSM*	MAXI*	MOTION*	ORIGIN*	PRETRM*	REBOUN*	RFORCE*
		SFORCE*	SUMARY*	SUMTAB*	TIRES*	TRIM*	TRUCK*		
CC	-	SETA*							
CE	-	TIRES*							
CG	-	AERO*	CGCALC*	GEOM*	INLOAD*	LAMBDA*	MOTION*	TRUCK*	
CGCALC	-	GEOM*							
CGMASS	-	BLDATA	CGCALC*	EXTRNL	FORCEI*	FSPRNG	GEOM*	LAMBDA	MASSM*
		MAXI	MOTION	REBOUN	RFORCE*	SFORCE*	TIRES	TRIM	TRUCK
CGPOS	-	BLDATA	CGCALC*	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA*	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
CGS	-	TIRES*							
CHECK	-	FSPRNG*	GEOM*	MOTION*	PRETRM*	TRIM*	TRUCK*		
CINT1	-	INT1*	TRUCK*						
CIP	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
CIQ	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
CK	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	INT1X*	MOTION	SETA*
		TIRES*	TRUCK						

CKK	-	SETA*							
CN	-	GEOM*	INPSPR	LAMBDA	TIRES*	TRIM	TRUCK		
CON	-	RELAX*							
COS	-	AERO*	BETAIJ*	DRAGL*	EULER*	SETA*	TIRES*		
COSA	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB
		TRUCK							
COSPHI	-	BETAIJ*	EULER*						
COSPSI	-	BETAIJ*							
COSTHE	-	BETAIJ*	EULER*						
COS1	-	AERO*							
COS2	-	AERO*							
COS3	-	AERO*							
CP	-	BLDATA	AERO*	DIFFL	DRAGL*	ENGULF	INLOAD	MOTION	SETA*
		TRUCK							
CPI	-	AERO*							
CPQ	-	HYTRUK*							
CPQT	-	HYTRUK*							
CPQT1	-	HYTRUK*							
CPQT2	-	HYTRUK*							
CPS	-	HYTRUK*							
CPSIG	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
CPST	-	HYTRUK*							
CPST1	-	HYTRUK*							
CPST2	-	HYTRUK*							
CPS0	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
CRIT	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY*	SUMTAB	TIRES	TRIM	TRUCK*		
CRKM	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
CT	-	GEOM*	INPSPR	LAMBDA	TIRES*	TRIM	TRUCK		
C1	-	HYTRUK*							
C2	-	HYTRUK*							
D	-	MAXI*	TIRES*						
DAMPF	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG*	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
DAMPV	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG*	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
DATAIN	-	BLDATA*	CGCALC*	EXTRNL*	FORCEI*	FSPRNG*	GEOM*	LAMBDA*	MASSM*
		MAXI*	MOTION*	REBOUN*	RFORCE*	SFORCE*	TIRES*	TRIM*	TRUCK*

DCG	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI*	MOTION*	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES	TRIM	TRUCK		
DCG0	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI*	MOTION	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY*	SUMTAB	TIRES	TRIM	TRUCK		
DD	-	MAXI*	TIRES*						
DDCODE	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
DDMIN	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI*	MOTION	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY*	SUMTAB	TIRES	TRIM	TRUCK		
DD1	-	MAXI*							
DD2	-	MAXI*							
DEFLEC	-	FSPRNG*							
DEL	-	TIRES*							
DELN	-	TIRES*							
DELTA	-	AERO	ATMOS*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA
		SUMTAB	TRUCK*						
DELTAS	-	BASE*	BOTTOM*	FSPRNG*	MAXI*	MOTION*	TRIM*	TRUCK*	
DELTAT	-	EULER*	ORIGIN*						
DELTD	-	TIRES*							
DELTIM	-	BLDATA	ADAMI*	ADAM5*	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*
		LAMBDA	MASSM	MAXI*	MOTION*	REBOUN	RFORCE	SFORCE	TIRES*
		TRIM	TRUCK						
DELTM	-	TIRES*							
DELTNX	-	TIRES	TRUCK*						
DELTN1	-	TIRES*	TRUCK*						
DELTT	-	TIRES*							
DELTX	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION*	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
DELTX1	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION*	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
DELX	-	RELAX*							
DET	-	RELAX*	SOLVE*						
DIF	-	TIRES*							
DIFFL	-	AERO*							
DIMENS	-	BLDATA*	BOTTOM*	FSPRNG*	GEOM*	INPSPR*	MASSM*	MOTION*	REBOUN*
		TRUCK*							
DISP	-	BASE	BOTTOM*	FSPRNG*	MAXI*	MOTION*	TRIM	TRUCK	

DISPJ	-	BASE*	PRETRM*						
DISX	-	TIRES*							
DISY	-	TIRES*							
DMAX	-	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK			
DMIN	-	ENGULF*	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK		
DP	-	DIFFL*	MAXI*						
DPRT	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM	LAMBDA	MASSM
		MAXI	MOTION*	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
DPRT1	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION*	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
DRAGL	-	AERO*	SETA*						
DS	-	ENGULF*							
DTTNX	-	TIRES	TRUCK*						
DTTN1	-	TIRES*	TRUCK*						
DW	-	TIRES*							
DVEL	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI*	MOTION*	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY*	SUMTAB*	TIRES	TRIM	TRUCK		
DX	-	MAXI*	RELAX*	TIRES*					
DZ	-	MAXI*	TIRES*						
D1	-	PITER*							
D2	-	PITER*							
E	-	TIRES*							
EE	-	SETA*							
EEF	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
EMAG	-	TIRES*							
ENDTIM	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION*	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
ENDTX	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION*	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
ENGULF	-	AERO*							
EPS	-	TIRES*							
ER	-	PRETRM*	RELAX*	TRIM*					
EU	-	TIRES*							
EUF	-	TIRES*							
EULC	-	BOTTOM*	EULER*	MOTION*	REBOUN*				
EULER	-	MOTION*							
EXP	-	ATMOS*	DIFFL*	HYTRUK*					
EXPAN	-	BOTTOM*	INPSPR*	TRIM	TRUCK				
EXTDIM	-	EXTRNL*							

EXTRNL	-	MOTION*	TRIM*	TRUCK*					
EX1	-	PITER*							
EX2	-	PITER*							
F	-	AERO*	FSPRNG*	PITER*					
FAERO	-	AERO*	BASE	BETAIJ	BOTTOM	EULER	EXTRNL*	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES	TRIM	TRUCK		
FDAMP	-	FSPRNG*							
FGRAV	-	EXTRNL*	GRAV*						
FI	-	EXTRNL*	FORCEI*	RFORCE*	SFORCE*				
FIPHIJ	-	RFORCE*	SFORCE*						
FIPSI	-	FORCEI*	RFORCE*	SFORCE*					
FITHET	-	RFORCE*							
FIXJ	-	RFORCE*							
FIYJ	-	FORCEI*	RFORCE*	SFORCE*					
FIZJ	-	RFORCE*	SFORCE*						
FLAG	-	GEOM*							
FLOAT	-	GEOM*	TRUCK*						
FMAX	-	PITER*							
FMIN	-	PITER*							
FN	-	TIRES*							
FND	-	TIRES*							
FOFX	-	BLDATA	BASE*	BOTTOM	FSPRNG*	INPSPR*	PRETRM*	TIRES*	TRIM*
		TRUCK							
FORB	-	TIRES*							
FORC	-	FSPRNG*							
FORCE	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL*	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION*	ORIGIN	PRETRM	REBOUN*	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES	TRIM*	TRUCK		
FORCEI	-	EXTRNL*							
FR	-	HYTRUK*	INT1X*						
FSPDIM	-	FSPRNG*							
FSPRNG	-	EXTRNL*							
FT	-	TIRES*							
FTO	-	TIRES*							
FTIRES	-	EXTRNL*							
FTMAX	-	TIRES*							
FWIRE	-	FSPRNG*							
F1	-	AERO*	PITER*	TRIM*					

F10	-	AERO*							
F2	-	AERO*	TRIM*						
F3	-	AERO*							
F4	-	AERO*							
F5	-	AERO*							
F6	-	AERO*							
F7	-	AERO*							
F8	-	AERO*							
F9	-	AERO*							
G	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
GDPHI	-	FORCEI*	RFORCE*	SFORCE*					
GDPSI	-	RFORCE*	SFORCE*						
GDTHET	-	FORCEI*	RFORCE*	SFORCE*					
GENACC	-	ADAM5*	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI
		FSPRNG	GEOM	MASSM	MAXI*	MOTION*	ORIGIN	PRETRM	REBOUN*
		RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM	TRUCK	
GENDIS	-	ADAM5*	AERO	BASE*	BETAIJ	BOTTOM*	EULER	EXTRNL	FORCEI
		FSPRNG*	GEOM	MASSM	MAXI*	MOTION*	ORIGIN	PRETRM*	REBOUN
		RFORCE	SFORCE	SUMARY	SUMTAB	TIRES*	TRIM*	TRUCK*	
GENDIX	-	TRUCK*							
GENSH	-	PRETRM*							
GENVEL	-	ADAM5*	AERO	BASE	BETAIJ	BOTTOM*	EULER*	EXTRNL	FORCEI*
		FSPRNG*	GEOM	MASSM	MAXI*	MOTION*	ORIGIN*	PRETRM	REBOUN*
		RFORCE*	SFORCE*	SUMARY	SUMTAB	TIRES*	TRIM*	TRUCK	
GEOM	-	TRUCK*							
GF	-	TIRES*							
GFCH	-	PRETRM*							
GFCH1	-	PRETRM*							
GFCH2	-	PRETRM*							
GFGCH	-	PRETRM*							
GFGCH1	-	PRETRM*							
GFGCH2	-	PRETRM*							
GFGH	-	PRETRM*							
GFGH1	-	PRETRM*							
GFSH	-	PRETRM*							
GFSH1	-	PRETRM*							
GFSH2	-	PRETRM*							
GMAT	-	TIRES*							

GMSQ	-	TIRES*							
GRAV	-	EXTRNL*							
GS	-	BLDATA	CGCALC	EXTRNL*	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES*	TRIM*	TRUCK
GSL	-	AERO*	EXTRNL*						
GVN	-	TIRES*							
GVNT	-	TIRES*							
GVT	-	TIRES*							
GVTD	-	TIRES*							
GVTT	-	TIRES*							
GXDOTJ	-	FORCEI*	RFORCE*	SFORCE*					
GYDOTJ	-	RFORCE*	SFORCE*						
GZDOTJ	-	RFORCE*	SFORCE*						
HPRIM	-	ATMOS*							
HPRIMB	-	ATMOS*							
HW	-	TIRES*							
HWK	-	TIRES*							
HYTRUK	-	AERO*							
I	-	ADAM5*	AERO*	BASE*	BOTTOM*	DIFFL*	ENGULF*	HYTRUK*	INLOAD*
		INT1*	INT1X*	MASSM*	MATXIN*	MAXI*	MULT*	PITER*	PRETRM*
		REBOUN*	RELAX*	SETA*	SOLVE*	TRUCK*			
IA	-	TIRES*	TRUCK*						
IB	-	MULT*							
IC	-	MATXIN*							
ID	-	TIRES*	TRUCK*						
IDD	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK*							
IDOF	-	LAMBDA*							
IDOFS	-	LAMBDA*							
IDUM	-	INPSPR*	LAMBDA*						
IDUM1	-	LAMBDA*							
IE	-	TIRES*							
IEND	-	INPSPR*	TRUCK*						
IFIRST	-	BOTTOM*	EXTRNL*	MOTION*	REBOUN	TIRES	TRUCK		
IFL	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
IH	-	BASE*	PRETRM*						
IHIT	-	BOTTOM*							

IHM4	-	BASE*	PRETRM*						
IHM5	-	BASE*	PRETRM*						
IHM6	-	BASE*	PRETRM*						
IHP1	-	BASE*	PRETRM*						
IHP2	-	BASE*	PRETRM*						
II	-	AERO*	TIRES*						
IIMAX	-	INT1*	TRUCK*						
IINT	-	INT1*	TRUCK*						
IIS	-	BLDATA	AERO*	DIFFL	ENGULF	INLOAD	MOTION	SETA	TRUCK
IIS3	-	AERO*							
IJ	-	TIRES*							
IK	-	GEOM*	MULT*	PRETRM*	TIRES*				
IL	-	BASE*							
IM	-	PRETRM*							
INDEX	-	ADAMI*	ADAM5*	BASE*	BOTTOM*	CGCALC*	EXTRNL*	FORCEI*	FSPRNG*
		GEOM*	GRAV*	INPSPR*	LAMBDA*	MASSM*	MATXIN*	MAXI*	MOTION*
		PRETRM*	RFORCE*	SUMARY*	TRIM*				
INDEXS	-	LAMBDA*							
INDEXY	-	EXTRNL*	FORCEI*	RFORCE*	SFORCE*				
INDEX1	-	ADAMI*	BASE*	FSPRNG*	GEOM*	GRAV*	INPSPR*	LAMBDA*	MASSM*
INDEX2	-	FSPRNG*	GEOM*	LAMBDA*					
INLOAD	-	AERO*							
INPSPR	-	GEOM*							
INTFLG	-	BOTTOM*	EXTRNL*	MOTION*	REBOUN*	TIRES*	TRUCK*		
INT1	-	BASE*	FSPRNG*	PRETRM*	TIRES*	TRIM*			
INT1X	-	HYTRUK*							
IO	-	CGCALC*	FSPRNG*	GEOM*	INPSPR*	LAMBDA*	MOTION*	TRIM*	TRUCK*
IOPT	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM*	GRAV	HYTRUK*	INPSPR	LAMBDA	MASSM	MAXI
		MOTION	PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES*
		TRIM	TRUCK						
IO1	-	CGCALC*	INPSPR*						
IP	-	RELAX*	SOLVE*						
IPOINT	-	INPSPR*							
IQRA	-	TIRES*							
IQYA	-	TIRES*							
IR	-	MULT*							
IRL	-	PRETRM*							
IRO	-	MULT*							

IROL	-	TIRES*	TRUCK*						
IRU	-	PRETRM*							
IS	-	BLDATA	AERO*	DIFFL	ENGULF	INLOAD	MATXIN*	MOTION	PRETRM*
		SETA	TRUCK						
ISD	-	FSPRNG*							
ISDAMP	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG*	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
ISK	-	PRETRM*							
ISKK	-	PRETRM*							
ISTART	-	BASE*	FSPRNG*	PRETRM*	TRIM*				
IT	-	MATXIN*	TIRES*	TRUCK*					
ITER	-	TRUCK*							
IU	-	BASE*							
J	-	ADAM5*	BASE*	CGCALC*	FORCEI*	FSPRNG*	GEOM*	HYTRUK*	INT1X*
		LAMBDA*	MASSM*	MATXIN*	MULT*	PRETRM*	RFORCE*	SFORCE*	SOLVE*
		SUMTAB*	TRUCK*						
JCURVE	-	BLDATA	BASE*	BOTTOM	FSPRNG*	INPSPR*	PRETRM*	TIRES*	TRIM*
		TRUCK							
JDEBUG	-	FSPRNG*	GEOM*	MOTION*	PRETRM*	TRIM*	TRUCK		
JF	-	AERO*							
JFIR	-	BLDATA	AERO*	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
JFTRIM	-	TIRES*	TRIM*	TRUCK*					
JGRAPH	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM*	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION*
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
JI	-	MULT*							
JJ	-	SUMTAB*							
JMAX	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
JP	-	TIRES*							
JPREL	-	BASE*							
JP1	-	TIRES*							
JTRIM	-	TIRES*	TRIM*	TRUCK					
J1	-	MASSM*							
J6	-	ADAM5*							
J7	-	ADAM5*							
K	-	AERO*	ATMOS*	DIFFL*	ENGULF*	FSPRNG*	INLOAD*	INPSPR*	INT1X*
		LAMBDA*	MATXIN*	MULT*	SETA*	SOLVE*	SUMTAB*	TRUCK*	
KAERO	-	AERO*							

KAXL8	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
KAXLES	-	ADAMI	ADAM5	AERO*	BASE*	BOTTOM*	CGCALC*	EXTRNL	FORCEI*
		FSPRNG*	GEOM*	GRAV*	INPSPR*	LAMBDA*	MASSM*	MAXI	MOTION
		PRETRM*	REBOUN*	RFORCE	SFORCE*	SUMARY	SUMTAB	TIRES*	TRIM*
		TRUCK*							
KB	-	SOLVE*							
KBL	-	HYTRUK*							
KBLAST	-	AERO	DIFFL	ENGULF	HYTRUK*	INLOAD	MOTION	SETA	SUMTAB
		TRUCK*							
KBOG	-	ADAMI	ADAM5	AERO	BASE	BOTTOM*	CGCALC	EXTRNL	FORCEI
		FSPRNG*	GEOM*	GRAV	INPSPR*	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM*	REBOUN*	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
KC1	-	PRETRM*							
KC2	-	PRETRM*							
KDAM	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION*
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB*	TIRES	TRIM
		TRUCK*							
KDEBUG	-	BLDATA	AERO*	DIFFL	ENGULF*	INLOAD*	MOTION	SETA*	TRUCK
KDOF	-	EXTRNL*	FORCEI*	MASSM*	RFORCE*	SFORCE*	TRIM*		
KDUM	-	MOTION*							
KERR	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL*	FORCEI
		FSPRNG	GEOM*	GRAV	INPSPR*	LAMBDA*	MASSM	MAXI	MOTION*
		PRETRM*	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB*	TIRES*	TRIM*
		TRUCK*							
KFLAG	-	SUMTAB*							
KG	-	GEOM*							
KGWIR1	-	ADAMI	ADAM5	AERO	BASE*	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM*	GRAV	INPSPR	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM*	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
KGWIR2	-	ADAMI	ADAM5	AERO	BASE*	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM*	GRAV	INPSPR	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM*	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							

KK	-	AERO*	SUMTAB*						
KMAX	-	MAXI	MOTION	SUMARY	SUMTAB*	TRUCK*			
KM1	-	SOLVE*							
KOFK	-	TIRES*							
KOK	-	PITER*	SUMTAB*	TRUCK*					
KOL	-	MATXIN*							
KP1	-	SOLVE*							
KRACKS	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC*	EXTRNL*	FORCEI
		FSPRNG*	GEOM*	GRAV*	INPSPR	LAMBDA*	MASSM*	MAXI	MOTION
		PRETRM*	REBOUN	RFORCE*	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
KRIGID	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM*	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
KS	-	LAMBDA*							
KSHELT	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC*	EXTRNL*	FORCEI
		FSPRNG*	GEOM*	GRAV*	INPSPR*	LAMBDA*	MASSM*	MAXI	MOTION
		PRETRM*	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
KSTART	-	ADAMI*	ADAM5*	BOTTOM*	EXTRNL*	MOTION*	REBOUN	TIRES	
KS1	-	LAMBDA*							
KS2	-	LAMBDA*							
KS3	-	LAMBDA*							
KS4	-	LAMBDA*							
KT	-	AERO*	INLOAD*						
KTIRE	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION*	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES*	TRIM*	TRUCK		
KTS	-	BLDATA	AERO*	DIFFL	ENGULF	INLOAD*	MOTION	SETA	TRUCK
KWIRES	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG*	GEOM*	GRAV	INPSPR*	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM*
		TRUCK							
KX	-	MASSM*	RFORCE*						
L	-	MATXIN*							
LAMBDA	-	GEOM*							
LL	-	MULT*							
LOAD	-	BLDATA*	AERO*	DIFFL*	ENGULF*	INLOAD*	MOTION*	SETA*	TRUCK*

M	-	ATMOS*	EXTRNL*	FORCEI*	GRAV*	INPSPR*	MASSM*	MATXIN*	MULT*
		REBOUN*	RFORCE*	SFORCE*	SOLVE*				
MASS	-	MASSM*	MOTION*	REBOUN*	TRUCK*				
MASSES	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC*	EXTRNL	FORCEI*
		FSPRNG	GEOM*	GRAV*	INPSPR	LAMBDA	MASSM*	MAXI*	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY*	SUMTAB	TIRES*	TRIM
		TRUCK							
MASSM	-	MOTION*							
MATDIM	-	MATXIN*							
MATXIN	-	MOTION*							
MAX	-	MATXIN*							
MAXDOF	-	BLDATA*	BOTTOM*	FSPRNG*	GEOM*	INPSPR	MASSM	MOTION*	REBOUN*
		TRUCK							
MAXI	-	MOTION*							
MAXMAS	-	BLDATA*	BOTTOM	FSPRNG	GEOM*	INPSPR	MASSM	MOTION	REBOUN
		TRUCK							
MAXMIN	-	MAXI*	MOTION*	SUMARY*	SUMTAB*	TRUCK*			
MAXSP	-	BLDATA*	BOTTOM*	FSPRNG*	GEOM*	INPSPR	MASSM	MOTION	REBOUN
		TRUCK							
MAXSPR	-	BLDATA*	BASE	BOTTOM	FSPRNG	INPSPR*	PRETRM	TIRES	TRIM
		TRUCK							
MAXZ	-	MATXIN*							
MDIMOP	-	BLDATA*	CGCALC	EXTRNL	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
MDOF	-	ADAMI*	ADAM5*	AERO*	BASE	BOTTOM*	CGCALC	EXTRNL*	FORCEI
		FSPRNG*	GEOM*	GRAV	INPSPR	LAMBDA*	MASSM*	MAXI*	MOTION*
		PRETRM*	REBOUN*	RFORCE	SFORCE	SUMARY*	SUMTAB	TIRES*	TRIM*
		TRUCK*							
MM	-	INPSPR*	MULT*	RELAX*					
MOTION	-	TRUCK*							
MOVING	-	EXTRNL*	MAXI*	MOTION*	ORIGIN*	TIRES*	TRIM*	TRUCK*	
MRDIM	-	BLDATA*	BOTTOM	FSPRNG	GEOM*	INPSPR	MASSM	MOTION	REBOUN
		TRUCK							
MSPRNG	-	ADAMI	ADAM5	AERO	BASE	BOTTOM*	CGCALC	EXTRNL	FORCEI
		FSPRNG*	GEOM*	GRAV	INPSPR	LAMBDA*	MASSM	MAXI*	MOTION*
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY*	SUMTAB	TIRES	TRIM
		TRUCK							
MULT	-	BOTTOM*	FSPRNG*	MOTION*	REBOUN*	TIRES*			
N	-	AERO*	BOTTOM*	DIFFL*	ENGULF*	FSPRNG*	INLOAD*	INPSPR*	LAMBDA*
		MATXIN*	MULT*	RELAX*	SETA*	SOLVE*	SUMTAB*	TRIM*	

NA	-	BOTTOM*	INPSPR*	REBOUN*					
NASPR	-	ADAMI	ADAM5	AERO	BASE	BOTTOM*	CGCALC	EXTRNL	FORCEI
		FSPRNG*	GEOM*	GRAV	INPSPR*	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM*	REBOUN*	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
NBOTOM	-	BOTTOM*	INPSPR	TRIM	TRUCK				
NBOX	-	BLDATA	AERO*	DIFFL	ENGULF*	INLOAD*	MOTION	SETA*	TRUCK
NCALL	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL*	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK*							
NCASE	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK*							
NCOUNT	-	PRETRM*	RELAX*	TRIM*					
NDET	-	SOLVE*							
NDIM	-	RELAX*	SOLVE*						
NDOF	-	LAMBDA*							
NDOFA	-	TIRES*							
NEQ	-	ADAMI*	ADAM5*	BOTTOM	EXTRNL	MOTION	REBOUN	RELAX*	TIRES
NFLAG	-	TRUCK*							
NIT	-	ADAMI*	ADAM5*	BOTTOM*	EXTRNL*	MOTION*	REBOUN	TIRES	
NJOB	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK*							
NK	-	TIRES*							
NM1	-	SOLVE*							
NN	-	MULT*							
NNQ	-	RELAX*							
NOK	-	PRETRM*	RELAX*	TRIM*					
NOR	-	TRUCK*							
NORMAX	-	TRUCK*							
NOUT	-	ADAMI	ADAM5	AERO*	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG*	GEOM	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION*
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
NP	-	BASE*	FSPRNG*	INPSPR*	PRETRM*	TRIM*			
NPDAMP	-	BLDATA	CGCALC	EXTRNL	FORCEI	FSPRNG*	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK

NPPC	-	BLDATA	BASE*	BOTTOM	FSPRNG*	INPSPR*	PRETRM*	TIRES*	TRIM*
		TRUCK							
NPR	-	PRETRM*	TRIM*						
NPRINT	-	ADAMI	ADAM5	AERO	BASE	BOTTOM*	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM*	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION*
		PRETRM	REBOUN	RELAX*	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES
		TRIM	TRUCK*						
NPS	-	BLDATA	AERO*	DIFFL	ENGULF*	INLOAD*	MOTION	SETA*	TRUCK
NPSS	-	AERO*	ENGULF*	INLOAD*	SETA*				
NPV	-	FSPRNG*							
NQ	-	RELAX*							
NRSPR	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM*	GRAV	INPSPR	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
NS	-	BOTTOM*	REBOUN*						
NSP	-	REBOUN*							
NSPRNG	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG*	GEOM*	GRAV	INPSPR*	LAMBDA	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
NSTOP	-	PRETRM*							
NT	-	HYTRUK*	LAMBDA*	TIRES*	TRUCK*				
NTIRES	-	GEOM*	INPSPR	LAMBDA*	TIRES*	TRIM*	TRUCK*		
NTMAX	-	TRUCK*							
NTRIAL	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA	MASSM	MAXI	MOTION
		PITER*	PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES
		TRIM	TRUCK*						
NTRY	-	PRETRM*	TRIM*						
NTSPRN	-	GEOM*	INPSPR*	LAMBDA	TIRES*	TRIM*	TRUCK		
NTT	-	LAMBDA*	TIRES*	TRUCK*					
NT1	-	HYTRUK*							
NT2	-	HYTRUK*							
NX	-	INT1*	INT1X*	TIRES*					
NX1	-	TIRES*							
NZ	-	MATXIN*							
OPTION	-	ADAMI*	ADAM5*	AERO*	BASE*	BOTTOM*	CGCALC*	EXTRNL*	FORCEI*
		FSPRNG*	GEOM*	GRAV*	INPSPR*	LAMBDA*	MASSM*	MAXI*	MOTION*
		PRETRM*	REBOUN*	RFORCE*	SFORCE*	SUMARY*	SUMTAB*	TIRES*	TRIM*
		TRUCK*							

ORIGIN	-	MOTION*							
P	-	AERO*	DIFFL*	HYTRUK*					
POSAV	-	BOTTOM	EULER*	MOTION*	REBOUN				
PI	-	BLDATA*	AERO*	DIFFL	ENGULF*	INLOAD	MOTION	SETA*	TRUCK*
PIMP	-	AERO	DIFFL	ENGULF	HYTRUK*	INLOAD	MOTION	SETA	SUMTAB*
		TRUCK*							
PITCH	-	AERO	BASE	BETAIJ*	BOTTOM	EULER*	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION*	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES	TRIM*	TRUCK*		
PITCHD	-	EULER*							
PITER	-	TRUCK*							
PI1	-	DRAGL*							
PL	-	HYTRUK*							
POSD	-	TIRES*							
PP	-	HYTRUK*							
PQ	-	HYTRUK*							
PQ1	-	HYTRUK*							
PR	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB
		TRUCK							
PRB	-	SETA*							
PRELOD	-	BASE*	GEOM*	PRETRM*	TRUCK*				
PRES	-	RELAX*							
PRETRM	-	TRIM*							
PROP	-	RELAX*							
PS	-	AERO*	DIFFL*	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB
		TRUCK							
PSAV	-	BOTTOM	EULER*	MOTION*	REBOUN				
PSINF	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
PSOEST	-	AERO	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB
		TRUCK*							
PST	-	HYTRUK*							
PST1	-	HYTRUK*							
PST2	-	HYTRUK*							
PS0	-	AERO*	DIFFL*	ENGULF	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB*
		TRUCK*							
PS00	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
PS1	-	HYTRUK*							
PTRIAL	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES	TRIM	TRUCK*		

PX	-	RELAX*							
P0	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB
		TRUCK*							
Q	-	AERO*	ATMOS*	HYTRUK*					
QIMP	-	AERO	DIFFL	ENGULF	HYTRUK*	INLOAD	MOTION	SETA	SUMTAB*
		TRUCK*							
QR	-	ADAMI	ADAM5*	BOTTOM	EXTRNL	MOTION	REBOUN	TIRES	
QRD	-	ADAMI*	ADAM5*	BOTTOM	EXTRNL	MOTION	REBOUN	TIRES	
QRP	-	ADAMI*	ADAM5*	BOTTOM	EXTRNL	MOTION	REBOUN	TIRES	
Q0	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB
		TRUCK*							
R	-	HYTRUK*	INT1*	MULT*	PITER*				
RANGE	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB
		TRUCK							
RANGEM	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB*
		TRUCK*							
RDSAV	-	BOTTOM	EULER*	MOTION*	REBOUN				
RE	-	ATMOS*							
REBOUN	-	BOTTOM*							
RELAX	-	PRETRM*	TRIM*						
RES	-	PRETRM*	RELAX*						
RETURN	-	ADAMI*	ADAM5*	AERO*	ATMOS*	BASE*	BETAIJ*	BOTTOM*	CGCALC*
		DIFFL*	DRAGL*	ENGULF*	EULER*	EXTRNL*	FORCEI*	FSPRNG*	GEOM*
		GRAV*	HYTRUK*	INLOAD*	INPSPR*	INT1*	INT1X*	LAMBDA*	MASSM*
		MATXIN*	MAXI*	MOTION*	MULT*	ORIGIN*	PITER*	PRETRM*	REBOUN*
		RELAX*	RFORCE*	SETA*	SFORCE*	SOLVE*	SUMARY*	SUMTAB*	TIRES*
		TRIM*							
RFORCE	-	EXTRNL*							
RHO	-	ATMOS*							
RHOZ	-	ATMOS*							
RHO0	-	AERO*							
RIR	-	MULT*							
RL	-	HYTRUK*							
RLP	-	RELAX*							
RLX	-	LAMBDA*							
RLY	-	LAMBDA*							
RLZ	-	LAMBDA*							
RNK	-	TIRES*							
ROLL	-	AERO	BASE	BETAIJ*	BOTTOM	EULER*	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION*	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES	TRIM*	TRUCK*		

ROLLD	-	EULER*							
ROW	-	MATXIN*							
RR	-	HYTRUK*							
RRES	-	RELAX*							
RSAV	-	BOTTOM	EULER*	MOTION*	REBOUN				
RT	-	HYTRUK*							
RT1	-	HYTRUK*							
RT2	-	HYTRUK*	PITER*						
RW	-	GEOM*	INPSPR	LAMBDA	TIRES*	TRIM	TRUCK		
R1	-	PITER*							
R3	-	PITER*							
S	-	BLDATA	AERO*	ATMOS*	DIFFL	ENGULF	INLOAD*	MOTION	SETA
		TRUCK							
SAVACC	-	BOTTOM*	MOTION*	TRUCK*					
SBOTM	-	BOTTOM*	INPSPR*	TRIM	TRUCK				
SE	-	TIRES*							
SETA	-	AERO*							
SF	-	AERO*	DIFFL	ENGULF	HYTRUK*	INLOAD	MOTION	PRETRM*	SETA
		SUMTAB	TRUCK						
SFORCE	-	EXTRNL*							
SG	-	AERO*							
SGS	-	TIRES*							
SIG	-	BLDATA	AERO*	DIFFL	ENGULF	INLOAD	MOTION	SETA*	TRUCK
SIGI	-	AERO*							
SIGMA	-	ATMOS*							
SIGMAB	-	ATMOS*							
SIGX	-	RELAX*							
SIN	-	AERO*	BETAIJ*	EULER*	SETA*	TIRES*	TRUCK*		
SINA	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB
		TRUCK							
SINPHI	-	BETAIJ*	EULER*						
SINPSI	-	BETAIJ*							
SINTHE	-	BETAIJ*	EULER*						
SI1	-	BLDATA	AERO	DIFFL	ENGULF	INLOAD*	MOTION	SETA*	TRUCK
SI2	-	BLDATA	AERO	DIFFL	ENGULF	INLOAD*	MOTION	SETA*	TRUCK
SI3	-	BLDATA	AERO	DIFFL	ENGULF	INLOAD*	MOTION	SETA*	TRUCK
SI4	-	BLDATA	AERO	DIFFL	ENGULF	INLOAD*	MOTION	SETA*	TRUCK
SJ1	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
SJ2	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK

SJ3	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
SJ4	-	BLDATA	AERO	DIFFL*	ENGULF	INLOAD	MOTION	SETA*	TRUCK
SLID	-	TIRES*	TRUCK*						
SLOPE	-	BLDATA	CGCALC	EXTRNL*	FORCEI	FSPRNG	GEOM*	LAMBDA	MASSM
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM*	TRUCK
SLX	-	LAMBDA*							
SLXA	-	LAMBDA*							
SLY	-	LAMBDA*							
SLZ	-	LAMBDA*							
SMAX	-	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK			
SMIN	-	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK			
SOLVE	-	RELAX*							
SPRING	-	BLDATA*	BASE*	BOTTOM*	FSPRNG*	INPSPR*	PRETRM*	TIRES*	TRIM*
		TRUCK*							
SPRNG	-	EXTRNL*							
SQRT	-	AERO*	ATMOS*	ENGULF*	LAMBDA*	MAXI*	SETA*	TIRES*	
STOP	-	TRUCK*							
SUMARY	-	TRUCK*							
SUMTAB	-	TRUCK*							
T	-	DIFFL*	ENGULF*	HYTRUK*	SOLVE*				
TA	-	ENGULF*							
TAMAX	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			
TAMIN	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			
TAN	-	ENGULF*	TIRES*						
TANA	-	ENGULF*							
TCGM	-	CGCALC*							
TCGX	-	CGCALC*							
TCGY	-	CGCALC*							
TCGZ	-	CGCALC*							
TD	-	BLDATA	AERO*	DIFFL	ENGULF*	HYTRUK*	INLOAD	MOTION	SETA
		TRUCK							
TDLAST	-	BLDATA	AERO*	DIFFL	ENGULF*	INLOAD	MOTION	SETA	TRUCK
TDMAX	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			
TDMIN	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			
TDT	-	HYTRUK*							
TDT1	-	HYTRUK*							
TDT2	-	HYTRUK*							
TERM1	-	TIRES*							

TGS	-	TIRES*							
THETA	-	ATMOS*							
TIM	-	AERO*							
TIME	-	ADAM5*	AERO*	BASE	BETAIJ	BOTTOM*	EULER	EXTRNL	FORCEI
		FSPRNG	GEOM	MASSM	MAXI*	MOTION*	ORIGIN	PRETRM	REBOUN
		RFORCE	SFORCE	SUMARY*	SUMTAB*	TIRES*	TRIM	TRUCK*	
TIM1	-	AERO*							
TI012	-	ADAMI*	ADAM5*	BOTTOM	EXTRNL	MOTION*	REBOUN	TIRES	
TI02	-	ADAMI*	ADAM5*	BOTTOM	EXTRNL	MOTION*	REBOUN	TIRES	
TI024	-	ADAMI*	ADAM5*	BOTTOM	EXTRNL	MOTION*	REBOUN	TIRES	
TI072	-	ADAMI*	ADAM5*	BOTTOM	EXTRNL	MOTION*	REBOUN	TIRES	
TIREC	-	GEOM*	INPSPR*	LAMBDA*	TIRES*	TRIM*	TRUCK*		
TIRES	-	EXTRNL*							
TIRSAV	-	TIRES*	TRUCK*						
TL	-	LAMBDA*							
TLX	-	GEOM	INPSPR	LAMBDA*	TIRES*	TRIM	TRUCK		
TLXB	-	TIRES*							
TLY	-	GEOM	INPSPR	LAMBDA*	TIRES*	TRIM*	TRUCK		
TLYB	-	LAMBDA*	TIRES*						
TLZ	-	GEOM	INPSPR	LAMBDA*	TIRES*	TRIM	TRUCK		
TLZB	-	LAMBDA*							
TM	-	ATMOS*							
TMB	-	ATMOS*							
TMIN	-	AERO	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI*	MOTION	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY*	SUMTAB*	TIRES	TRIM	TRUCK		
TMZ	-	ATMOS*							
TOL1	-	PITER*							
TOL2	-	PITER*							
TOTAL	-	TRIM*							
TRIM	-	TRUCK*							
TRMDIM	-	TRIM*							
TS	-	HYTRUK*	TRIM*						
TSMAX	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			
TSMIN	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			
TSQ	-	HYTRUK*							
TT	-	HYTRUK*							
TVMAX	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			
TVMIN	-	MAXI*	MOTION	SUMARY*	SUMTAB	TRUCK			

T0	-	AERO*							
T1	-	DIFFL*	ENGULF*						
T2	-	DIFFL*							
T3	-	DIFFL*							
T4	-	DIFFL*							
U	-	GEOM*	INPSPR	LAMBDA	TIRES*	TRIM	TRUCK		
V	-	DIFFL*	FSPRNG*						
VB	-	TIRES*							
VDAMP	-	FSPRNG*							
VDOM	-	REBOUN*							
VDT	-	TIRES*							
VEL	-	BOTTOM*	INPSPR	TRIM	TRUCK				
VIMP	-	REBOUN*							
VMAX	-	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK			
VMIN	-	MAXI*	MOTION*	SUMARY*	SUMTAB	TRUCK			
VNUM	-	REBOUN*							
VS	-	AERO*	DIFFL	ENGULF*	HYTRUK	INLOAD	MOTION	SETA*	SUMTAB
		TRUCK							
VT	-	TIRES*							
W	-	AERO*	DIFFL	ENGULF	HYTRUK	INLOAD	MOTION	SETA	SUMTAB*
		TRUCK*							
WEIGHT	-	GEOM*	GRAV*	PRETRM*	TRIM*	TRUCK*			
WF	-	SETA*							
WG	-	BLDATA	AERO	DIFFL	ENGULF	INLOAD	MOTION	SETA*	TRUCK
WGTS	-	GEOM*	GRAV*	PRETRM*	TRIM	TRUCK			
WW	-	SETA*							
X	-	BLDATA	BASE*	BOTTOM	FSPRNG*	HYTRUK*	INPSPR*	INT1*	INT1X*
		PRETRM*	RELAX*	TIRES*	TRIM*	TRUCK			
XB	-	INLOAD*							
XBARCM	-	AERO*	BASE	BETAIJ	BOTTOM	EULER	EXTRNL	FORCEI*	FSPRNG
		GEOM*	MASSM*	MAXI	MOTION	ORIGIN	PRETRM*	REBOUN	RFORCE*
		SFORCE*	SUMARY	SUMTAB	TIRES*	TRIM	TRUCK		
XBOGIE	-	ADAMI	ADAM5	AERO	BASE	BOTTOM	CGCALC	EXTRNL	FORCEI
		FSPRNG	GEOM	GRAV	INPSPR	LAMBDA*	MASSM	MAXI	MOTION
		PRETRM	REBOUN	RFORCE	SFORCE	SUMARY	SUMTAB	TIRES	TRIM
		TRUCK							
XBOX	-	BLDATA	AERO*	DIFFL	ENGULF*	INLOAD*	MOTION	SETA	TRUCK
XCG	-	CGCALC*	GEOM*	INLOAD*	LAMBDA*	MOTION	TRUCK		
XDSAVE	-	EXTRNL	MAXI	MOTION*	ORIGIN*	TIRES	TRIM	TRUCK	

XI	-	TIRES*							
XIYZ	-	BLDATA	CGCALC	EXTRNL	FORCEI*	FSPRNG	GEOM*	LAMBDA	MASSM*
		MAXI	MOTION	REBOUN	RFORCE	SFORCE	TIRES	TRIM	TRUCK
XI1	-	BLDATA	CGCALC	EXTRNL	FORCEI*	FSPRNG	GEOM*	LAMBDA	MASSM*
		MAXI	MOTION	REBOUN	RFORCE*	SFORCE*	TIRES	TRIM	TRUCK
XI2	-	BLDATA	CGCALC	EXTRNL	FORCEI*	FSPRNG	GEOM*	LAMBDA	MASSM*
		MAXI	MOTION	REBOUN	RFORCE*	SFORCE*	TIRES	TRIM	TRUCK
XI3	-	BLDATA	CGCALC	EXTRNL	FORCEI*	FSPRNG	GEOM*	LAMBDA	MASSM*
		MAXI	MOTION	REBOUN	RFORCE*	SFORCE*	TIRES	TRIM	TRUCK
XLAMDA	-	BASE*	BOTTOM*	EXTRNL	FSPRNG*	LAMBDA*	PRETRM*	REBOUN*	TRIM
		TRUCK							
XLJ	-	FORCEI*	RFORCE*	SFORCE*					
XMASS	-	MASSM*	MOTION*	REBOUN*	TRUCK				
XMJ	-	FORCEI*	RFORCE*	SFORCE*					
XOBAR	-	EXTRNL*	MAXI*	MOTION*	ORIGIN*	TIRES*	TRIM*	TRUCK*	
XOBARD	-	ORIGIN*							
XPREL	-	BASE*							
XPROP	-	RELAX*							
XRES	-	RELAX*							
XSAVE	-	EXTRNL	MAXI	MOTION	ORIGIN*	TIRES	TRIM*	TRUCK*	
XSAVEX	-	TRUCK*							
XSC	-	TIRES*							
XT	-	INT1*	INT1X*						
XTB	-	EXTRNL	MAXI*	MOTION	ORIGIN	TIRES*	TRIM	TRUCK	
XTBAR	-	TIRES*							
XX1	-	RELAX*							
X0	-	ENGULF*							
X1	-	AERO*							
X3	-	AERO*							
X4	-	AERO*							
YAW	-	AERO	BASE	BETAIJ*	BOTTOM	EULER*	EXTRNL	FORCEI	FSPRNG
		GEOM	MASSM	MAXI	MOTION*	ORIGIN	PRETRM	REBOUN	RFORCE
		SFORCE	SUMARY	SUMTAB	TIRES	TRIM*	TRUCK*		
YAWD	-	EULER*							
YB	-	INLOAD*							
YBOX	-	BLDATA	AERO*	DIFFL	ENGULF	INLOAD*	MOTION	SETA	TRUCK
YCG	-	CGCALC*	GEOM*	INLOAD*	LAMBDA*	MOTION	TRUCK		
YDSAV	-	BOTTOM	EULER*	MOTION*	REBOUN				

```
YDSAVE   -  EXTRNL   MAXI     MOTION*  ORIGIN*  TIRES    TRIM     TRUCK
YIELD    -  MAXI     MOTION   SUMARY   SUMTAE*  TRUCK*
YLJ      -  FORCEI*  RFORCE*  SFORCE*
YOBAR    -  EXTRNL   MAXI     MOTION*  ORIGIN*  TIRES*   TRIM*    TRUCK*
YOBARD   -  ORIGIN*
YSAV     -  BOTTOM   EULER*   MOTION*  REBOUN
YSAVE    -  EXTRNL   MAXI     MOTION   ORIGIN*  TIRES    TRIM*    TRUCK*
YSAVEX   -  TRUCK*
YT       -  INT1*
YTBAR    -  TIRES*
Z        -  AERO*    ATMOS*   DIFFL    ENGULF   HYTRUK   INLOAD   MOTION   SETA*
            SUMTAB   TRUCK
ZB       -  INLOAD*
ZBOX     -  BLDATA   AERO*    DIFFL    ENGULF*  INLOAD*  MOTION   SETA     TRUCK
ZCG      -  CGCALC*  GEOM*    INLOAD*  LAMBDA*  MOTION   TRUCK
ZDSAVE   -  EXTRNL   MAXI     MOTION*  ORIGIN*  TIRES    TRIM     TRUCK
ZLJ      -  FORCEI*  RFORCE*  SFORCE*
ZOBAR    -  EXTRNL   MAXI*    MOTION*  ORIGIN*  TIRES*   TRIM*    TRUCK*
ZOBARD   -  ORIGIN*
ZSAVE    -  EXTRNL   MAXI     MOTION   ORIGIN*  TIRES    TRIM*    TRUCK*
ZSAVEX   -  TRUCK*
ZTB      -  EXTRNL   MAXI*    MOTION   ORIGIN   TIRES*   TRIM     TRUCK
ZTBAR    -  TIRES*

NO UNUSED VARIABLES APPEAR IN THE ABOVE LIST.
```

APPENDIX B

PROGRAM LISTING

```
C
C     MAIN PROGRAM.
C
C     A DYNAMIC MATHEMATICAL MODEL OF A TRUCK-SHELTER-RACK
C     SYSTEM IN RESPONSE TO A BLAST WAVE.
C
C
C     FILE USE -
C
C     TAPE 5 = INPUT
C     TAPE 6 = OUTPUT
C     TAPE 8 = DATA TAPE FOR GRAPHICS.
C
C
C     JULY, 1977, VERSION 2.0.
C
C     KERR -  0, NO ERROR
C             1, NEW ORIENTATION
C             2, NEW JOB
C             3, ABORT JOB.
C     NCALL - 0, RESPONSE
C             1, TRIM
C             2, SETUP.
C     KTIRE - 0, FIRST PASS
C             1, NEW TRIM
C             2, OLD TRIM
C
      COMMON/BLAST/  A,AD,AD1,AD2,ALT,A0,COSA,DELTA,KBLAST,PIMP,PR,
     1   PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/BOTM/   EXPAN(100),NBOTOM(100),SBOTM(100),VEL(100)
      COMMON/CALC/   BETA (3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/CG/     XCG,YCG,ZCG
      COMMON/CHECK/  JDEBUG
      COMMON/CINT1/  IIMAX,IINT
      COMMON/DATAIN/ CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DELTAS/ DISP(100)
      COMMON/DIMENS/ MAKDOF,MAXMAS,MAXSP,MRDIM
      COMMON/INTFLG/ IFIRST
      COMMON/ITER/   GENDIX(50),XSAVEX,YSAVEX,ZSAVEX
      COMMON/JFTRIM/ JTRIM
      COMMON KLAMDA(100,44)
      COMMON/LOAD/   AL(6),A6(6),CK(6),CP(6),DDCODE(6),
     1   EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2   NPS(6,4),PI,PSINF(6),PS00(6),S(16,6,4),SIG(5),
     3   SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,6,4),
     4   SJ1(16,6,4),SJ2(16,6,4),SJ3(16,6,4),SJ4(16,6,4),
     5   TD(16,6,4),WG(6),XBOX(16,6,4),YBOX(16,6,4),
     6   ZBOX(16,6,4), TDLAST
      COMMON/MASS/   XMASS(50,50)
      COMMON/MAXMIN/ AMAX(50),AMIN(50),ANGLE(19),CIP(19,6),
     1   CIQ(19,6),CPSIQ(19,6),CPS0(19,6),CRKM(19,6),DMAX(50),
     2   DMIN(50),IFL(19,6),JMAX,KMAX,SMAX(100),SMIN(100),
     3   TAMAX(50),TAMIN(50),TDMAX(50),TDMIN(50),TSMAX(100),
     4   TSMIN(100),TVMAX(50),TVMIN(50),VMAX(50),VMIN(50),
     5   YIELD(6)
      COMMON/MOVING/ XDSAVE,XOBAR,XSAVE,XTB(5),YDSAVE,YOBAR,YSAVE,
     1   ZDSAVE,ZOBAR,ZSAVE,ZTB(5)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
      COMMON/PRELOD/ PRELOD(4)
      COMMON/SPRING/ FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1   X(600)
      COMMON /TIREC/ CN,CT,NTIRES(6),NTSPRN,RW,TLX(3,6),TLY(6),TLZ(6),U
      COMMON /TIRSAV/ DELTNX(3,2,18),DELTN1(3,2,18),DTTNX(2,18),
     1  DTTN1(2,18),SLID(2,18)
      COMMON/WEIGHT/ WEIGHT,WGTS(13)
C
```

TRUCK

```
      DATA IEND/2HEN/
C
      NTMAK = 10
      NJOB = 0
  100 NJOB = NJOB + 1
      READ (5,1) (IDD(I),I=1,40)
      IF(IDD(1).EQ.IEND) GO TO 900
      WRITE (6,20) NJOB
      WRITE (6,21) (IDD(I),I=1,40)
      KERR = 0
      NCALL = 2
      CALL GEOM
      IF (KERR.GT.0) GO TO 800
      CALL AERO(0,0.0)
      WRITE (6,3)
      NCALL = 1
      TIME = -1.
      IINT = 0
      IIMAK = 20
      CALL TRIM
      IT = 0
      DO 130 IA = 1,KAXLES
      NTT = NTIRES(IA)
      DO 130 NT = 1,NTT
      IT = IT + 1
      DO 130 IROL = 1,2
      DTTNK(IROL,IT) = DTTN1(IROL,IT)
      DO 130 ID=1,3
  130 DELTNX(ID,IROL,IT) = DELTN1(ID,IROL,IT)
      XSAVEX = XSAVE
      YSAVEX = YSAVE
      ZSAVEX = ZSAVE
      DO 150 IO = 1,MDOF
  150 GENDIX(IO) = GENDIS(IO)
      READ (5,4) KDAM,KBLAST
      IF (KBLAST.EQ.0) KBLAST = 1
      IF (KBLAST.EQ.1) WRITE (6,15)
      IF (KBLAST.EQ.2) WRITE (6,16)
      IF (KDAM.EQ.0) WRITE (6,8)
      IF (KDAM.NE.0) WRITE (6,9)
      CALL SUMTAB (0,0)
      NCASE = 0
C
  200 READ (5,5) PSOEST,AD1,AD2,DELTA,W,ALT
      IF (PSOEST.EQ.0.0) GO TO 750
      WRITE (6,6) PSOEST,AD1,AD2,DELTA,W,ALT
      AD = 0.
      PS0 = PSOEST
      CALL AERO(1,0.0)
      A00 = A0/12.
      WRITE (6,7) P0,A00
      NORMAX = 1
      IF (DELTA.NE.0.0) NORMAX = (AD2-AD1)/DELTA + 1.001
      IF (NORMAX.LE.0) NORMAX = 1
C
      DO 700 NOR = 1,NORMAX
      NCASE = NCASE + 1
      AD = AD1 + FLOAT(NOR-1)*DELTA
      KERR = 0
      IF (NOR.GT.1) PS0 = 0.95*PS0*(1.0 + .35*SIN(AD*PI/180.)**2)/
     1 (1.0 + .35*SIN((AD-DELTA)*PI/180.)**2)
      NFLAG = 0
      IF (KDAM.GT.0) CALL SUMTAB(1,0)
      NTRIAL = 0
  400 NTRIAL = NTRIAL + 1
      IT = 0
      DO 430 IA = 1,KAXLES
      NTT = NTIRES(IA)
      DO 430 NT = 1,NTT
      IT = IT + 1
      DO 430 IROL = 1,2
      DTTN1(IROL,IT) = DTTNX(IROL,IT)
      SLID(IROL,IT) = 0.0
```

```
      DO 430 ID=1,3
  430 DELTN1(ID,IROL,IT) = DELTNX(ID,IROL,IT)
      XSAVE = XSAVEX
      YSAVE = YSAVEX
      ZSAVE = ZSAVEX
      XOBAR = XSAVEX
      YOBAR = YSAVEX
      ZOBAR = ZSAVEX
      DO 500 IO = 1,MDOF
  500 GENDIS(IO) = GENDIX(IO)
      ROLL = GENDIS(1)
      PITCH = GENDIS(4)
      YAW = GENDIS(5)
      IINT = 0
      IIMAX = 20
      IF (NPRINT.EQ.0) IIMAX = -1
      NCALL = 1
      TIME = -1.
      CALL BETAIJ
      CALL EXTRNL
      TIME = 0.
      NCALL = 0
      CALL AERO (1,0.0)
      WRITE (6,10) NJOB,NCASE,NTRIAL,AD,PS0,Q0,RANGEM,PIMP,QIMP
      CALL MOTION
      IF (KERR.EQ.2) GO TO 750
      CALL SUMARY
      IF (KERR.EQ.1) GO TO 700
      IF (KDAM.EQ.0)  GO TO 700
      PTRIAL(1) = PS0
      CALL PITER (CRIT,PTRIAL,NTRIAL,KOK)
      PS0 = PTRIAL(1)
      CALL AERO(1,0.0)
      IF (PS0.EQ.PTRIAL(1)) GO TO 600
      PTRIAL(1) = PS0
      IF (PS0.EQ.PTRIAL(2).OR.PS0.EQ.PTRIAL(3)) NFLAG = 1
      NFLAG = NFLAG + 1
      IF (NFLAG.EQ.1) GO TO 500
      KOK = 2
      WRITE (6,11) PS0,RANGEM,CRIT(1)
  600 IF (KOK.EQ.0 .AND.NTRIAL.LT.NTMAX) GO TO 400
      IF (NTRIAL.EQ.NTMAX) WRITE (6,13) (PTRIAL(I),CRIT(I),I=1,5)
      IF (KOK.EQ.1) WRITE (6,12) NJOB,NCASE,AD,PS0,Q0,RANGEM,PIMP,QIMP
      CALL SUMTAB(2,KOK)
  700 CONTINUE
      GO TO 200
C
  750 IF (JMAX.EQ.0) GO TO 100
      WRITE (6,17) NJOB,(IDD(I),I=1,40),(YIELD(K),K=1,KMAX)
      DO 760 J=1,JMAX
      WRITE (6,18) ANGLE(J),(CPS0(J,K),IFL(J,K),K=1,KMAX)
      WRITE (6,19) (CRKM(J,K),K=1,KMAX)
      WRITE (6,19) (CIQ(J,K),K=1,KMAX)
      WRITE (6,19) (CPSIQ(J,K),K=1,KMAX)
      WRITE (6,19) (CIP(J,K),K=1,KMAX)
  760 CONTINUE
      GO TO 100
C
C
C       END OF RUN.
C
  800 WRITE (6,14)
      STOP
C
  900 WRITE (6,2)
      STOP
C
C       FORMAT STATEMENTS
C
    1 FORMAT (40A2)
    2 FORMAT (1H0,56X,21H*** NORMAL END OF JOB)
    3 FORMAT (25H0START TRIM CALCULATIONS.)
    4 FORMAT (6I12)
```

```
  5 FORMAT (6F12.1)
  6 FORMAT (11H0BLAST DATA/
   1    29H     ESTIMATED PS0, PSI          = E15.6/
   2    29H     AZIMUTHAL ANGLE 1, DEG      = E15.6/
   3    29H     AZIMUTHAL ANGLE 2, DEG      = E15.6/
   5    29H     INCREMENT IN ANGLE, DEG = E15.6/
   5    29H     YIELD, KT                   = E15.6/
   7    29H     ALTITUDE, FT                = E15.6)
  7 FORMAT (19H0AMBIENT CONDITIONS/
   1    28H     PRESSURE, PSI           = E15.6/
   2    28H     SPEED OF SOUND, FT/SEC = E15.6)
  8 FORMAT (18H0RESPONSE RUN ONLY)
  9 FORMAT (14H0ITERATION RUN)
 10 FORMAT (5H1JOB I2,4X,5HCASE I2,4X,6HTRIAL I2/
   1    39H     AZIMUTHAL ANGLE, DEG                = E15.6/
   2    39H     PEAK OVERPRESSURE, PSI              = E15.6/
   3    39H     PEAK DYNAMIC PRESSURE, PSI          = E15.6/
   4    39H     RANGE, METERS                       = E15.6/
   5 39H    OVERPRESSURE IMPULSE, PSI-SEC          = E15.6/
   5    39H     DYNAMIC PRESSURE IMPULSE, PSI-SEC = E15.6)
 11 FORMAT (63H0ITERATION TERMINATED BECAUSE ADDITIONAL BLAST DATA IS
   1REQUIRED/27H     LIMITING OVERPRESSURE = E15.6/
   2    25H     CORRESPONDING RANGE = E15.6/
   4    24H     CORRESPONDING CRIT = E15.6)
 12 FORMAT (37H0FINAL ESTIMATED RESULTS OF ITERATION/
   1    7H0     JOB I2,4X,5HCASE I2/
   1    39H     AZIMUTHAL ANGLE, DEG                = E15.6/
   2    39H     PEAK OVERPRESSURE, PSI              = E15.6/
   3    39H     PEAK DYNAMIC PRESSURE, PSI          = E15.6/
   4    39H     RANGE, METERS                       = E15.6/
   5 39H    OVERPRESSURE IMPULSE, PSI-SEC          = E15.6/
   5    39H     DYNAMIC PRESSURE IMPULSE, PSI-SEC = E15.6)
 13 FORMAT (10H0MORE THAN, I3,16H  TRIALS REQIRED/
   1    4X, 8HPRESSURE,8X,4HCRIT/(2E15.6))
 14 FORMAT (1H0,56X,23H*** ABNORMAL END OF JOB)
 15 FORMAT (23H0BLAST MODEL - 1 KT, SL)
 16 FORMAT (26H0BLAST MODEL - 60 W**1/3 M)
 17 FORMAT (16H1SUMMARY OF JOB I3/1X,40A2//
   1    39H THE FIVE CRITICAL VALUES TABULATED ARE/
   2    29H     (1) PEAK OVERPRESSURE, PSI/24H     (2) RANGE, KILOMETERS/
   3    40H     (3) DYNAMIC PRESSURE IMPULSE, PSI-SEC/
   4    33H     (4) THE PRODUCT OF (1) AND (3)/
   5    36H     (5) OVERPRESSURE IMPULSE, PSI-SEC//
   5    48H AN ASTERISK (*) INDICATES A QUESTIONABLE RESULT//26X,
   7    10HYIELD (KT)/1H ,3X,9HAZIMUTHAL /1H ,3X,11HANGLE (DEG),
   8    (6F11.2))
 18 FORMAT (1H0,F10.2,6X,6(F9.3,A2))
 19 FORMAT (1H ,14X,6F11.3)
 20 FORMAT (1H1,40X,38HGENERALIZED TRUCK SHELTER, VERSION 2.0/54X,
   1    4HJOB I3//)
 21 FORMAT (1H0,1X,9HRUN ID = ,40A2,/)
    END
```

TRUCK

```
      SUBROUTINE ADAMI(DELTIM)
C
C     INITIALIZE INTEGRATION VARIABLES FOR OPEN ADAMS METHOD.
C
      COMMON/ADAM/    KSTART,NEQ,NIT,QR(100),QRD(6,100),QRP(100),TI012,
     1   TI02,TI024,TI072
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
C
      NEQ = MDOF*2
      NIT=1
      KSTART=1
      TI02=DELTIM/2.0
      TI012=DELTIM/12.0
      TI024=DELTIM/24.0
      TI072=DELTIM/720.0
      DO 10 INDEX=1,NEQ
      QRP(INDEX)=0.0
      DO 10 INDEX1=1,6
   10 QRD(INDEX1,INDEX)=0.0
C
      RETURN
C
      END
```

ADAMI

```
      SUBROUTINE ADAM5(GENDIS,GENVEL,GENACC,TIME,DELTIM)
C
C     FIFTH ORDER OPEN ADAMS INTEGRATION.
C
      COMMON/ADAM/    KSTART,NEQ,NIT,QR(100),QRD(6,100),QRP(100),TIO12,
     1   TIO2,TIO24,TIO72
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,K3OG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IOD(40)
C
      DIMENSION GENDIS( 1),GENVEL( 1),GENACC( 1)
C
C     INITIALIZE.
C
      DO 500 INDEX=1,MDOF
      QR(INDEX)=GENDIS(INDEX)
      QR(INDEX+MDOF)=GENVEL(INDEX)
      QRD(1,INDEX)=GENVEL(INDEX)
  500 QRD(1,INDEX+MDOF)=GENACC(INDEX)
C
C     INTEGRATE.
C
      GO TO (3000,3001,3006,3006), NIT
 3000 TIME = TIME+DELTIM
 3001 IF (KSTART.EQ.1) GO TO 3027
C
      DO 3003 I=1,NEQ
      QRP(I)=QR(I)
C
      DO 3003 J=1,5
      J6=6-J
      J7=7-J
 3003 QRD(J7,I)=QRD(J6,I)
C
      GO TO (3004,3004,3011,3015,3019,3023), KSTART
 3004 DO 3005 I=1,NEQ
 3005 QR(I)=QRP(I)+QRD(2,I)*DELTIM
C
      NIT=3
      TIME=TIME+DELTIM
      GO TO 3025
C
 3006 GO TO (3007,3007,3013,3017,3021),KSTART
 3007 DO 3008 I=1,NEQ
 3008 QR(I)=QRP(I)+(QRD(1,I)+QRD(2,I))*TIO2
C
 3009 NIT=NIT+1
      IF (NIT.NE.5) GO TO 3025
      KSTART=KSTART+1
      NIT=1
      GO TO 3025
C
 3011 DO 3012 I=1,NEQ
 3012 QR(I)=QRP(I)+(3.0*QRD(2,I)-QRD(3,I))*TIO2
      NIT=3
      GO TO 3025
C
 3013 DO 3014 I=1,NEQ
 3014 QR(I)=QRP(I)+(5.0*QRD(1,I)+8.0*QRD(2,I)-QRD(3,I))*TIO12
      GO TO 3009
C
 3015 DO 3016 I=1,NEQ
 3016 QR(I)=QRP(I)+(23.0*QRD(2,I)-16.0*QRD(3,I)+5.0*QRD(4,I))*TIO12
      NIT=3
      GO TO 3025
C
 3017 DO 3018 I=1,NEQ
 3018 QR(I)=QRP(I)+(9.0*QRD(1,I)+19.0*QRD(2,I)-5.0*QRD(3,I)+QRD(4,I))*
     1         TIO24
      GO TO 3009
C
 3019 DO 3020 I=1,NEQ
```

```
 3020 QR(I)=QRP(I)+(55.0*QRD(2,I)-59.0*QRD(3,I)+37.0*QRD(4,I)-
     1          9.0*QRD(5,I))*TID24
      NIT=3
      GO TO 3025
C
 3021 DO 3022 I=1,NEQ
 3022 QR(I)=QRP(I)+(251.0*QRD(1,I)+646.0*QRD(2,I)-264.0*QRD(3,I)+
     1           106.0*QRD(4,I)-19.0*QRD(5,I))*TI072
      GO TO 3009
C
C     ONCE INTEGRATION HAS STARTED, THIS EQUATION IS ALWAYS USED.
C     NIT=1, KSTART=6.
C
C
 3023 DO 3024 I=1,NEQ
 3024 QR(I)=QRP(I)+(1901.0*QRD(2,I)-2774.0*QRD(3,I)+
     1          2616.0*QRD(4,I)-1274.0*QRD(5,I)+251.0*QRD(6,I))*TI072
      GO TO 3025
C
C
 3027 KSTART=2
      NIT=2
      TIME=TIME-DELTIM
 3025 DO 5000 INDEX=1,MDOF
      GENDIS(INDEX)=QR(INDEX)
 5000 GENVEL(INDEX)=QR(INDEX+MDOF)
C
      RETURN
C
      END
```

ADAM5

```
      SUBROUTINE AERO (KAERO,GSL)
C     COMPUTES LOADS ON TRUCK - SHELTER.
C     KAERO - 0, INPUT
C             1, NEW BLAST DATA
C             2, RESPONSE.
      COMMON/BLAST/  A,AD,AD1,AD2,ALT,A0,COSA,DELTA,KBLAST,PIMP,PR,
     1   PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/CALC/   BETA (3,3),CRIT(5),DCG,DCGO,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/LOAD/    AL(6),A6(6),CK(6),CP(6),DDCODE(5),
     1   EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2   NPS(6,4),PI,PSINF(6),PS00(6),S(16,6,4),SIG(5),
     3   SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,6,4),
     4   SJ1(16,6,4),SJ2(16,6,4),SJ3(16,6,4),SJ4(16,6,4),
     5   TD(16,6,4),WG(5),XBOX(16,6,4),YBOX(16,6,4),
     6   ZBOX(16,6,4), TDLAST
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
C
      IF (KAERO-1) 100,150,180
C     FIRST PASS.
  100 CALL INLOAD
      IS = 2 + KAXLES
      IIS = 7 + 2*KAXLES
      IIS3 = IIS + 3
      RETURN
C
  150 CALL ATMOS (ALT,X1,SF(4),RHO0,X3,P0,A0,X4,KK)
      SF(3) = P0
      P0 = 14.696*P0
      SF(5) = A0/1116.45
      SF(1) = (W/SF(3))**(1./3.)
      SF(2) = SF(1)/SF(5)
      A0 = A0*12.0
      CALL HYTRUK (T0,RANGE,PS0,Q0,3)
      RANGEM = RANGE*0.3048
      Z = PS0/P0
      PR = 2.*(4.*Z+7)/(Z+7.)
      VS = A0*SQRT(1.0 + 6.0*Z/7.0)
      CALL SETA
      CALL ENGULF
      TIM1 = -1.0
      Q = 0.
      RETURN
C
  180 DO 200 I=1,MDOF
  200 FAERO(I) = 0.0
C     BY-PASS AERO AFTER POSITIVE PHASE.
      IF (TIME.GT.T0+TDLAST) RETURN
      CG = COS(GSL)
      SG = SIN(GSL)
      COS1 = COSA*(BETA(1,1)*CG + BETA(2,1)*SG) + SINA*BETA(3,1)
      COS3 = COSA*(BETA(1,2)*CG + BETA(2,2)*SG) + SINA*BETA(3,2)
      COS2 = COSA*(BETA(1,3)*CG + BETA(2,3)*SG) + SINA*BETA(3,3)
      AL(1) = ACOS(COS1)
      AL(3) = ACOS(COS2)
      AL(5) = ACOS(COS3)
      AL(2) = PI-AL(1)
      AL(4) = PI-AL(3)
      AL(6) = PI-AL(5)
      IF (KDEBUG.EQ.2.AND.NOUT.EQ.0) WRITE (6,4) AL
C     COMPUTE CP.
      DO 220 I=1,6
      CALL DRAGL (AL(I),CP(I))
  220 CONTINUE
      F1 = 0.
      F2 = 0.
      F3 = 0.
      F4 = 0.
      F5 = 0.
```

```
      F6 = 0.
      F7 = 0.
      F8 = 0.
      F9 = 0.
      F10= 0.
      DO 1300 N=1,NBOX
      KT = KTS(N)
      DO 1100 I=1,6
      NPSS = NPS(I,N)
      IF (NPSS.EQ.0) GO TO 1100
      CPI = CP(I)
      SIGI = SIG(I)
      II = (I+1)/2
      DO 1000 K=1,NPSS
      P = 0.
      JF = JFIR(K,I,N)
      TIM = TIME - TD(K,I,N)
      IF (TIM.LT.0.) GO TO 300
      IF (TIM.NE.TIM1) CALL HYTRUK (TIM,RANGE,PS,Q,2)
      TIM1 = TIM
      P = PS + CPI*Q
      IF (JF.EQ.1) GO TO 300
      CALL DIFFL (K,I,N,TIM,P)
  300 F = SIGI *P*S(K,I,N)
      IF (KDEBUG.EQ.2.AND.NOUT.EQ.0) WRITE (6,1) K,I,N,JFIR(K,I,N),
     1  TIM,P,Q,F
      IF (II-2) 350,550,450
  350 F1 = F1 - F*YBOX(K,I,N)
      F2 = F2 + F
      F5 = F5 + F*ZBOX(K,I,N)
      IF (KT.EQ.2) F7 = F7 - F*(YBOX(K,I,N) - XBARCM(2,IS))
      GO TO 1000
  450 F1 = F1 + F*XBOX(K,I,N)
      F3 = F3 + F
      F4 = F4 - F*ZBOX(K,I,N)
      IF (KT.EQ.1) GO TO 1000
      F7 = F7 + F*(XBOX(K,I,N) - XBARCM(1,IS))
      F8 = F8 + F
      F9 = F9 - F*(ZBOX(K,I,N) - XBARCM(3,IS))
      GO TO 1000
  550 F4 = F4 + F*YBOX(K,I,N)
      F5 = F5 - F*XBOX(K,I,N)
      F6 = F6 + F
      IF (KT.EQ.1) GO TO 1000
      F9 = F9 + F*(YBOX(K,I,N) - XBARCM(2,IS))
      F10 = F10 + F
 1000 CONTINUE
 1100 CONTINUE
 1300 CONTINUE
      FAERO(1) = F1
      FAERO(2) = F2
      FAERO(3) = F3
      FAERO(4) = F4
      FAERO(5) = F5
      FAERO(6) = F6
      FAERO(IIS) = F7
      FAERO(IIS+1) = F8
      FAERO(IIS+2) = F9
      FAERO(IIS+3) = F10
      IF (KDEBUG.EQ.2.AND.NOUT.EQ.0) WRITE (6,2) (FAERO(I),
     1  I=1,6),(FAERO(I),I=IIS,IIS3)
      RETURN
C
    1 FORMAT (4I3,4E15.6)
    2 FORMAT (35H GENERALIZED FORCES - TRUCK,SHELTER/(10E13.5))
    4 FORMAT (3H0AL/6E15.6/19H K,I,N,JF,TIM,P,Q,F)
      END
```

AERO

```
      SUBROUTINE ATMOS(Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)
C     CALLING SEQUENCE
C     CALL ATMOS(Z,TM,SIGMA,RHO,THETA,DELTA,CA,AMU,K)
C     Z      = GEOMETRIC ALTITUDE (FT)
C     TM     = MOLECULAR SCALE TEMPERATURE (DEGREES RANKINE)
C     SIGMA  = RATIO OF DENSITY TO THAT AT SEA LEVEL
C     RHO    = DENSITY (SLUGS/FT**3)
C     THETA  = RATIO OF TEMPERATURE TO THAT AT SEA LEVEL
C     DELTA  = RATIO OF PRESSURE TO THAT AT SEA LEVEL
C     CA     = SPEED OF SOUND (FT/SEC)
C     AMU    = VISCOSITY COEFFICIENT (LB-SEC/FT**2)
C     K = 1  NORMAL,
C       = 2  ALTITUDE GREATER THAN 300000. FT.,
C       = 3  ALTITUDE NEGATIVE,

      DIMENSION HPRIMB(11),TMB(11),SIGMAB(11),ALM(11)
      DATA HPRIMB/0.0,36089.239,82020.997,154199.480,173884.510,
     1259186.350,295275.590,344488.190,524934.380,557742.780,656167.980/
     2,TMB/518.688,389.988,389.988,508.788,508.788,298.188,298.188,
     3406.188,2386.188,2566.188,2836.188/,SIGMAB/1.00,2.9706958E-01,
     43.2665751E-02,1.2117870E-03,5.8677311E-04,1.7329156E-05,
     51.7928595E-06,9.3921519E-08,7.7658593E-10,5.6324877E-10,
     62.5726771E-10/,ALM/-0.00356616,0.0,0.00164592,0.0,-0.00246888,0.0,
     70.00219456,0.01097280,0.00548640,0.00274320,0.00192024/
      DATA Q/0.018744176/,RE/2.0855531E07/,S/198.72/,
     1AMUZ/3.7372998E-07/,RHOZ/0.0023769/,TMZ/518.688/
      K=1
      IF (Z) 1,3,2
    1 K=3
      GO TO 11
    2 IF(Z.GT.300000.)  K=K+1
    3 HPRIM=(RE/(RE+Z))*Z
      DO 4 M=1,11
      IF(HPRIM-HPRIMB(M)) 5,6,4
    4 CONTINUE
      M=12
    5 M=M-1
    6 IF(ALM(M)) 7,8,7
    7 TM=TMB(M)+ALM(M)*(HPRIM-HPRIMB(M))
      SIGMA=EXP((1.0+(Q/ALM(M)))*(ALOG(TMB(M)/TM)))*SIGMAB(M)
      GO TO 9
    8 TM=TMB(M)
      SIGMA=SIGMAB(M)*EXP(-(Q*(HPRIM-HPRIMB(M)))/TMB(M))
    9 RHO=RHOZ*SIGMA
      THETA=TM/TMZ
      DELTA=SIGMA*THETA
      CA=49.02177*SQRT(TM)
      AMU=AMUZ*SQRT(THETA**3)*((TMZ+S)/(TM+S))
   11 RETURN
      END
```

ATMOS

```
      SUBROUTINE BASE
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DELTAS/ DISP(100)
      COMMON XLAMDA(100,44)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IOO(40)
      COMMON/PRELOD/ PRELOD(4)
      COMMON/SPRING/ FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1    X(600)
C
C     DO WE HAVE ANY GUY WIRES?
C
      IF (KGWIR1.EQ.0) RETURN
C
C     SHIFT SPRINGS.
C
C
C     IH = HEAVE DOF OF SHELTER.
C
      IH=8+2*KAXLES
C
C     IHP1 = PITCH DOF.
C
      IHP1=IH+1
C
C
C     IHP2 = Z DOF
C
      IHP2 = IH + 2
C     IHM6 = PITCH INDEX FOR LAMBDA MATRIX.
C
      IHM6=IH-6
C
C     IHM5 = HEAVE INDEX FOR LAMBDA MATRIX.
C
      IHM5=IH-5
C
C     IHM4 = Z INDEX FOR LAMBDA MATRIX
C
      IHM4 = IH - 4
      DO 100 INDEX=KGWIR1,KGWIR2
      J=2*INDEX-1
      DISPJ=XLAMDA(J     ,IHM6)*GENDIS(IH)+XLAMDA(J     ,IHM5)*GENDIS(IHP1
     1) + XLAMDA(J,IHM4)*GENDIS(IHP2)
      ISTART=JCURVE(INDEX)
      NP=NPPC(INDEX)
      IL=ISTART
      IU=IL+NP-1
      DO 20 I=IL,IU
   20 FOFX(I)=-FOFX(I)
      JPREL=1
      CALL INT1(PRELOD(JPREL),NP,FOFX(ISTART),X(ISTART),XPREL)
      DO 30 I=IL,IU
   30 FOFX(I)=-FOFX(I)
      JPREL=JPREL+1
      DO 50 INDEX1=1,NP
      X(ISTART)=X(ISTART)-XPREL+DISPJ
   50 ISTART=ISTART+1
  100 CONTINUE
      RETURN
      END
```

BASE

```
      SUBROUTINE BETAIJ
C
C     BETA MATRIX IN TERMS OF EULER ANGLES.
C
C
C     SUBROUTINE BETAIJ CALCULATES THE TRANSFORMATION MATRIX FROM
C          THE MOVING SYSTEM TO THE INERTIAL SYSTEM, BETA.
C
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
C
      COSTHE=COS(YAW)
      SINTHE=SIN(YAW)
      COSPHI=COS(PITCH)
      SINPHI=SIN(PITCH)
      COSPSI=COS(ROLL)
      SINPSI=SIN(ROLL)
C
      BETA(1,1) = COSTHE*COSPSI
      BETA(2,1) = COSTHE*SINPSI
      BETA(3,1) = -SINTHE
      BETA(1,2) = COSPSI*SINTHE*SINPHI-SINPSI*COSPHI
      BETA(2,2) = COSPSI*COSPHI+SINPSI*SINTHE*SINPHI
      BETA(3,2) = COSTHE*SINPHI
      BETA(1,3) = SINPSI*SINPHI+COSPSI*SINTHE*COSPHI
      BETA(2,3) = SINPSI*SINTHE*COSPHI-COSPSI*SINPHI
      BETA(3,3) = COSTHE*COSPHI
C
      RETURN
C
      END
```

BETAIJ

```
      BLOCK DATA
C
      COMMON/DATAIN/ CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DIMENS/ MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/LOAD/    AL(6),A6(6),CK(6),CP(6),DDCODE(5),
     1   EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2   NPS(6,4),PI,PSINF(6),PS00(6),S(16,6,4),SIG(5),
     3   SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,5,4),
     4   SJ1(16,6,4),SJ2(16,5,4),SJ3(16,6,4),SJ4(15,5,4),
     5   TD(16,6,4),WG(5),XBOX(16,6,4),YBOX(16,6,4),
     6   ZBOX(16,6,4), TDLAST
      COMMON/SPRING/ FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1   X(600)
C
      DATA PI/3.14159265358 98/
      DATA MAXSP/100/, MAXSPR/600/, MAXDOF/50/, MRDIM/3/, MAXMAS/10/,
     1 MDIMDP/600/
      END
```

BLOCK DATA

```
      SUBROUTINE BOTTOM
C
C     SUBROUTINE BOTTOM IDENTIFIES ALL SPRINGS THAT
C          HAVE "BOTTOMED - OUT"
C
      COMMON/ADAM/    KSTART,NEQ,NIT,QR(100),QRD(5,100),QRP(100),TI012,
     1   TI02,TI024,TI072
      COMMON/BOTM/    EXPAN(100),NBOTOM(100),SBOTM(100),VEL(100)
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DELTAS/ DISP(100)
      COMMON/DIMENS/ MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/EULC/    PDSAV,PSAV,RDSAV,RSAV,YDSAV,YSAV
      COMMON/INTFLG/ IFIRST
      COMMON XLAMDA(100,44)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
      COMMON/SPRING/ FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1   X(600)
C
C     STEP 1 - TRANSFORM GENERALIZED DISPLACEMENTS
C              INTO SPRING DEFLECTIONS.
C
      CALL MULT (XLAMDA,GENDIS(7),DISP,MAXSP,MAXDOF-6,MSPRNG,MDOF-6,1)
      CALL MULT (XLAMDA,GENVEL(7),VEL,MAXSP,MAXDOF-6,MSPRNG,MDOF-6,1)
      IF (KBOG.EQ.0) GO TO 80
      NA=0
      DO 20 I=1,KAXLES
   20 NA=NA+NASPR(I)
      NS=2*NA+1
      DO 40 I=1,KBOG
      DISP(NS)=(DISP(NS)+DISP(NS+2))/2.0
      DISP(NS+1)=(DISP(NS+1)+DISP(NS+3))/2.0
      DISP(NS+2)=0.0
      DISP(NS+3)=0.0
      VEL(NS) = (VEL(NS) + VEL(NS+2))/2.0
      VEL(NS+1) = (VEL(NS+1) + VEL(NS+3))/2.
      NS=NS+4
   40 CONTINUE
C
C     STEP 2 - CHECK FOR MAX. DEFLECTION.
C
   80 IHIT=0
      DO 100 INDEX=1,MSPRNG
C
C     SBOTM = MINIMUM VALUE OF X ON FORCE DEFLECTION CURVES.
C
      IF (DISP(INDEX).GT.SBOTM(INDEX).AND.DISP(INDEX).LT.EXPAN(INDEX))
     1          GO TO 100
      IF (DISP(INDEX).LE.SBOTM(INDEX).AND.VEL(INDEX).GE.0.0) GO TO 100
      IF (DISP(INDEX).GE.EXPAN(INDEX).AND.VEL(INDEX).LE.0.0) GO TO 100
      IHIT=IHIT+1
      NBOTOM(IHIT)=INDEX
  100 CONTINUE
C
C     NO PROBLEM IF IHIT=0.
C
      IF (IHIT.EQ.0) RETURN
      DO 200 INDEX=1,IHIT
      N=NBOTOM(INDEX)
      IF (NPRINT.GT.0) WRITE (6,900) TIME,N,DISP(N)
  900 FORMAT (11H AT TIME = E13.6, 9H  SPRING I2,
     1   28H  BOTTOMS OUT, DEFLECTION = E13.6)
      CALL REBOUN(N)
  200 CONTINUE
C     SET UP RESTART PROGRAM.
      NIT = 2
      KSTART = 2
      IFIRST = 0
      RETURN
```

END

BOTTOM

```
      SUBROUTINE CGCALC
C
C     CALCULATE SYSTEM C.G.
C
      COMMON/CG/      XCG,YCG,ZCG
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
C
      WRITE (6,1)
C
C     ZERO OUT VARIABLES FOR C.G. CALCULATION.
C
      TCGM=0.0
      TCGX=0.0
      TCGY=0.0
      TCGZ=0.0
C
C     C.G. CALCULATION.
C
      DO 122 IO=1,MASSES
      TCGM=TCGM+CGMASS(IO)
      TCGX=TCGX+CGMASS(IO)*CGPOS(1,IO)
      TCGY=TCGY+CGMASS(IO)*CGPOS(2,IO)
      TCGZ=TCGZ+CGMASS(IO)*CGPOS(3,IO)
  122 WRITE (6,2) IO,CGMASS(IO),(CGPOS(IO1,IO),IO1=1,3)
C
C     ADD IN RHS RACKS.
C
      IF (KRACKS.EQ.0) GO TO 126
      INDEX=2+KAXLES
      IF (KSHELT.GT.0) INDEX=INDEX+1
      J=INDEX+KRACKS
      DO 124 IO=1,KRACKS
      TCGM=TCGM+CGMASS(INDEX)
      TCGX=TCGX-CGMASS(INDEX)*CGPOS(1,INDEX)
      TCGY=TCGY+CGMASS(INDEX)*CGPOS(2,INDEX)
      TCGZ=TCGZ+CGMASS(INDEX)*CGPOS(3,INDEX)
      CGMASS(J)=CGMASS(INDEX)
      CGPOS(1,J)=-CGPOS(1,INDEX)
      CGPOS(2,J)=CGPOS(2,INDEX)
      CGPOS(3,J)=CGPOS(3,INDEX)
      J=J+1
  124 INDEX=INDEX+1
  126 CONTINUE
C
C     LET USER KNOW NEW C.G.
C
      XCG=TCGX/TCGM
      YCG=TCGY/TCGM
      ZCG=TCGZ/TCGM
      WRITE (6,3) TCGM,XCG,YCG,ZCG
C
      RETURN
C
C     FORMAT STATEMENTS.
C
    1 FORMAT (6H0INDEX,12X,4HMASS,15X,1HX,15X,1HY,15X,1HZ,/)
    2 FORMAT (I6,5X,4E16.7)
    3 FORMAT (17H0SYSTEM CG MASS =,E16.7,
     1   4H X =,E16.7,4H Y =,E16.7,4H Z =,E16.7/)
C
      END
```

CGCALC

```
      SUBROUTINE DIFFL (K,I,N,T,P)
C     DIFFRACTION LOADING ON CLOSED, RECTANGULAR STRUCTURE.
C     T IS TIME FROM WHEN POINT IS FIRST ENGULFED.
C     P IS DRAG PHASE PRESSURE ON INPUT - DIFFRACTION PRESSURE OUT.
C
      COMMON/BLAST/  A,AD,AD1,AD2,ALT,A0,COSA,DELTA,<BLAST,PIMP,PR,
     1    PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/LOAD/    AL(6),AS(6),CK(6),CP(6),DDCODE(5),
     1    EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2    NPS(6,4),PI,PSINF(6),PS00(6),S(16,6,4),SIG(5),
     3    SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,6,4),
     4    SJ1(16,6,4),SJ2(16,6,4),SJ3(16,6,4),SJ4(16,6,4),
     5    TD(16,6,4),WG(6),XBOX(16,6,4),YBOX(16,6,4),
     5    ZBOX(16,6,4), TDLAST
      T1 = (AMAX1(T/SJ1(K,I,N),1.0) - 1.0)*EEF(1,I)
      T2 = (AMAX1(T/SJ2(K,I,N),1.0) - 1.0)*EEF(2,I)
      T3 = (AMAX1(T/SJ3(K,I,N),1.0) - 1.0)*EEF(3,I)
      T4 = (AMAX1(T/SJ4(K,I,N),1.0) - 1.0)*EEF(4,I)
      V = CK(I)*(T1 + T2 + T3 + T4)
      V = EXP(-V)
      DP = PSINF(I) + (PS00(I)-PSINF(I))*V
      DP = DP*PS/PS0
      IF (DDCODE(I).EQ.2) GO TO 300
      IF (P.LT.DP) GO TO 200
  100 JFIR(K,I,N) = 1
      RETURN
  200 P = DP
      RETURN
  300 IF (P.GT.DP) GO TO 200
      GO TO 100
C
      END
```

DIFFL

```
      SUBROUTINE DRAGL (ALPHA,CP)
C     DRAG PHASE LOADING ON CLOSED, RECTANGULAR STRUCTURE.
      DATA PI1/1.832595715/
      IF (ALPHA.LT.PI1) GO TO 100
      CP = -0.4
      GO TO 200
  100 CP = 0.85*COS(1.125*ALPHA)
  200 RETURN
      END
```

DRAGL

```
      SUBROUTINE ENGULF
C     COMPUTE ENGULFMENT TIMES FOR ALL PARTS OF SHELTER.
      COMMON/BLAST/  A,AD,AD1,AD2,ALT,AO,COSA,DELTA,KBLAST,PIMP,PR,
     1   PS,PSOEST,PSO,PO,QIMP,QO,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/LOAD/   AL(6),A6(6),CK(6),CP(6),DDCODE(5),
     1   EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2   NPS(6,4),PI,PSINF(6),PSOO(6),S(16,6,4),SIG(5),
     3   SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,6,4),
     4   SJ1(16,6,4),SJ2(16,6,4),SJ3(16,6,4),SJ4(16,6,4),
     5   TD(16,6,4),WG(6),XBOX(16,6,4),YBOX(16,6,4),
     6   ZBOX(16,6,4), TDLAST
C
      TDLAST = 0.
      IF (A.GT.PI/2. - 1.E-6) GO TO 400
      IF (A.LT.-PI/2. + 1.E-6) GO TO 1000
      X0 = XBOX(1,1,1) + 1000.
      TANA = TAN(A)
      T1 = 1./(TANA**2 + 1.0)
      DMIN = 1.E10
      DO 200 N=1,NBOX
      DO 200 I=1,6
      NPSS = NPS(I,N)
      IF (NPSS.EQ.0) GO TO 200
      DO 100 K=1,NPSS
      T = ((X0 - XBOX(K,I,N))*TANA + ZBOX(K,I,N))*T1
      DS = (XBOX(K,I,N) - X0 + T*TANA)**2 + (ZBOX(K,I,N) - T)**2
      TD(K,I,N) = DS
      IF (DS.LT.DMIN) DMIN = DS
  100 CONTINUE
  200 CONTINUE
      DMIN = SQRT(DMIN)
      TA = DMIN/VS
      DO 350 N=1,NBOX
      DO 350 I=1,6
      NPSS = NPS(I,N)
      IF (NPSS.EQ.0) GO TO 350
      DO 300 K=1,NPSS
      TD(K,I,N) = SQRT(TD(K,I,N))/VS - TA
      IF (TD(K,I,N).GT.TDLAST) TDLAST = TD(K,I,N)
      IF (KDEBUG.EQ.0) GO TO 300
      IF (TD(K,I,N).EQ.0.0) WRITE (6,2000) N,I,K
  300 CONTINUE
  350 CONTINUE
      GO TO 900
C
C     FRONT OF TRUCK.
C
  400 DMIN = -1.E10
      DO 500 N=1,NBOX
      IF (NPS(3,N).EQ.0) GO TO 500
      IF (ZBOX(1,3,N).GT.DMIN) DMIN = ZBOX(1,3,N)
  500 CONTINUE
  600 DO 800 N=1,NBOX
      DO 800 I=1,6
      NPSS = NPS(I,N)
      IF (NPSS.EQ.0) GO TO 800
      DO 700 K=1,NPSS
      TD(K,I,N) = ABS(ZBOX(K,I,N) - DMIN)/VS
      IF (TD(K,I,N).GT.TDLAST) TDLAST = TD(K,I,N)
      IF (KDEBUG.EQ.0) GO TO 700
      IF(TD(K,I,N).EQ.0.0)  WRITE(6,2000) N,I,K
  700 CONTINUE
  800 CONTINUE
  900 IF (KDEBUG.EQ.0) RETURN
      WRITE (6,2100)
      DO 950 N=1,NBOX
      DO 950 I=1,6
      NPSS = NPS(I,N)
      IF (NPSS.EQ.0) GO TO 950
      WRITE (6,2200) N,I,(TD(K,I,N),K=1,NPSS)
  950 CONTINUE
      RETURN
C
```

```
C       REAR OF TRUCK.
C
 1000 DMIN = 1.E10
      DO 1100 N = 1,NBOX
      IF (NPS(4,N).EQ.0) GO TO 1100
      IF (ZBOX(1,4,N).LT.DMIN) DMIN = ZBOX(1,4,N)
 1100 CONTINUE
      GO TO 600
C
 2000 FORMAT (32H POINT INTERCEPTED AT TIME ZERO 3I4)
 2100 FORMAT (12H0DELAY TIMES)
 2200 FORMAT (2I3,8E15.6/6X,3E15.6)
      END
```

ENGULF

```
      SUBROUTINE EULER(DELTAT)
C
C     COMPUTE EULER ANGLES.
C
C
C     DELTAT = TIME STEP  INCREMENT.
C
C     OUTPUT -
C
C     ROLL   = EULER ROLL ANGLE.
C     PITCH  = EULER PITCH ANGLE.
C     YAW    = EULER YAW ANGLE.
C     ROLLD  = EULER  ROLL RATE. (D-ROLL/D-T)
C     PITCHD = EULER  PITCH RATE.  (D-PITCH/D-T)
C     YAWD   = EULER  YAW RATE. (D-YAW/D-T)
C
      COMMON/CALC/    BETA (3,3),CRIT(5),DCG,DCG0,DOMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/EULC/    PDSAV,PSAV,RDSAV,RSAV,YDSAV,YSAV
C
      COSTHE=COS(YSAV)
      SINTHE=SIN(YSAV)
      COSPHI=COS(PSAV)
      SINPHI=SIN(PSAV)
C
C     ROLL, PITCH, YAW RATES.
C
      ROLLD=(GENVEL(1)*COSPHI+GENVEL(5)*SINPHI)/COSTHE
      PITCHD=GENVEL(4)+ROLLD*SINTHE
      YAWD=GENVEL(5)*COSPHI-GENVEL(1)*SINPHI
C
C     COMPUTE ROLL, PITCH, YAW EULER ANGLES.
C
      ROLL=RSAV +0.5*(RDSAV +ROLLD)*DELTAT
      PITCH=PSAV +0.5*(PDSAV +PITCHD)*DELTAT
      YAW=YSAV +0.5*(YDSAV +YAWD)*DELTAT
C
      RSAV = ROLL
      PSAV = PITCH
      YSAV = YAW
      RDSAV =ROLLD
      PDSAV =PITCHD
      YDSAV =YAWD
C
      RETURN
      END
```

EULER

```
      SUBROUTINE EXTRNL
C
C
C     SUBROUTINE EXTRNL COMPUTES THE EXTERNAL FORCES, Q.
C
      COMMON/ADAM/    KSTART,NEQ,NIT,QR(100),QRD(6,100),QRP(100),TIO12,
     1   TIO2,TIO24,TIO72
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/ CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MOIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/EXTDIM/ FGRAV(50),FI(50),FTIRES(50),SPRNG(50)
      COMMON/INTFLG/ IFIRST
      COMMON XLAMDA(100,44)
      COMMON/MOVING/ XDSAVE,XOBAR,XSAVE,XTB(5),YDSAVE,YOBAR,YSAVE,
     1   ZDSAVE,ZOBAR,ZSAVE,ZTB(5)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
C
      DO 20 INDEX=1,MDOF
   20 SPRNG(INDEX)=0.0
C
C     GRAVITATIONAL FORCE.
C
      CALL GRAV(BETA,FGRAV)
C
C     SPRING FORCES.
C
      IF (MDOF.GT.6) CALL FSPRNG(SPRNG)
      IF (NCALL.GT.0) GO TO 22
      IF (NIT.EQ.2) GO TO 22
      IF (NIT.EQ.3) GO TO 22
      IF (KSTART.EQ.6.AND.IFIRST.EQ.1) GO TO 22
      GO TO 50
   22 CONTINUE
C
C     TIRE FORCES.
C
      CALL TIRES(SLOPE,FTIRES)
      IF (KERR.GT.0) RETURN
C
C     ARE WE STILL IN TRIM?
C
      IF (NCALL.EQ.0) GO TO 50
      DO 25 INDEX=1,MDOF
   25 FORCE(INDEX)=-FGRAV(INDEX)+SPRNG(INDEX)+FTIRES(INDEX)
      RETURN
C
C     AERODYNAMIC FORCE.
C
   50 GSL = GS(1)
      IF (KOBAR.GT.SLOPE) GSL = GS(2)
      CALL AERO(2,GSL)
C
      M=0
      KDOF=0
      INDEXY=0
      CALL FORCEI(FI,M,KDOF,INDEXY)
      IF (KSHELT.EQ.0) GO TO 60
      CALL SFORCE(FI,M,KDOF,INDEXY)
   60 CONTINUE
      IF (KRACKS.EQ.0) GO TO 80
      CALL RFORCE(FI,M,KDOF,INDEXY)
   80 CONTINUE
C
C     TOTAL FORCE.
C
      DO 100 INDEX=1,MDOF
  100 FORCE(INDEX)=-FGRAV(INDEX)+SPRNG(INDEX)+FAERO(INDEX)-FI(INDEX)+
```

EXTRNL

```
      1              FTIRES(INDEX)
C
      RETURN
C
      END
```

EXTRNL

```
      SUBROUTINE FORCEI(FI,M,KDOF,INDEXY)
C
C     FORCEI CALCULATES NON-ACCELERATION INERTIA FORCES FROM
C        AXLE MASSES.
C
      COMMON/CALC/    BETA (3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/ CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1    DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2    XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IDD(40)
      DIMENSION FI(1)
C
      FIYJ(XMJ,XLJ,YLJ,ZLJ,GXDOTJ) = XMJ*(GENVEL(1)*GENVEL(2)-YLJ
     1                              *(GENVEL(1)**2+GENVEL(4)**2)
     2                              +GENVEL(5)*(XLJ*GENVEL(4)+ZLJ
     3                              *GENVEL(1))+2.0*GENVEL(1)*GXDOTJ)
     4                              -GENVEL(4)*GENVEL(6)*XMJ
      FIPSI(J,GDPHI,GDTHET) = (XI2(J) -XI1(J) )*GENVEL(4)
     1                        *GENVEL(5)-(XI1(J) +XI3(J) -XI2(J)
     2                        )*GENVEL(5)*GDPHI
     3                        +(XI2(J) +XI3(J) -XI1(J))*
     4                        GENVEL(4)*GDTHET
C
      M=1
      FI(1) = (XI2(1) -XI1(1) +CGMASS(1)*(XBARCM(1,1)**2
     1        -XBARCM(2,1)**2))*GENVEL(4)*GENVEL(5)-(XIYZ
     2        +CGMASS(1)*XBARCM(2,1)*XBARCM(3,1))*GENVEL(1)*GENVEL(4)
      FI(2) = -GENVEL(1)*GENVEL(3)*CGMASS(1)
     1        +GENVEL(5)*GENVEL(6)*CGMASS(M)
      FI(3) = GENVEL(1)*GENVEL(2)*CGMASS(1)
     1        -GENVEL(4)*GENVEL(6)*CGMASS(M)
      FI(4) = (XIYZ +CGMASS(1)*XBARCM(2,1)*XBARCM(3,1))
     1        *(GENVEL(1)**2-GENVEL(5)**2)+(XI3(1) -XI2(1)
     2        +CGMASS(1)*(XBARCM(2,1)**2-XBARCM(3,1)**2))*
     3        GENVEL(1)*GENVEL(5)
      FI(5) = (XI1(1) -XI3(1) +CGMASS(1)*(XBARCM(3,1)**2
     1        -XBARCM(1,1)**2))*GENVEL(1)*GENVEL(4)+(XIYZ
     2        +CGMASS(1)*XBARCM(2,1)*XBARCM(3,1))*GENVEL(4)*GENVEL(5)
      FI(6) = GENVEL(4)*GENVEL(3)*CGMASS(1)-GENVEL(5)*GENVEL(2)
     1        *CGMASS(1)
      IF (MASSES.EQ.1) GO TO 200
C
C     AXLE CONTRIBUTION.
C
      M=2
      DO 100 INDEX=1,KAXLES
      FI(1)=FI(1)+(XI2(M) -XI1(M) +CGMASS(M)
     1        *(XBARCM(1,M)**2-XBARCM(2,M)**2))*GENVEL(4)*GENVEL(5)
     2        -CGMASS(M)*XBARCM(2,M)*XBARCM(3,M)*GENVEL(1)*GENVEL(4)
     3        +2.0*GENVEL(1)*CGMASS(M)*XBARCM(2,M)*GENVEL(6+2*INDEX)
      FI(2) = FI(2)-GENVEL(1)*(GENVEL(3)*CGMASS(M)+2.0*CGMASS(M)
     1        *GENVEL(5+2*INDEX))
     2        +GENVEL(5)*GENVEL(6)*CGMASS(M)
      FI(3) = FI(3)+GENVEL(1)*GENVEL(2)*CGMASS(M)
     1        -GENVEL(4)*GENVEL(6)*CGMASS(M)
      FI(4) = FI(4)+CGMASS(M)*XBARCM(2,M)*XBARCM(3,M)*(GENVEL(1)**2
     1        -GENVEL(5)**2)+(XI3(M) -XI2(M) +CGMASS(M)
     2        *(XBARCM(2,M)**2-XBARCM(3,M)**2 ))*GENVEL(1)*GENVEL(5)
     3        +2.0*GENVEL(4)*CGMASS(M)*XBARCM(2,M)*GENVEL(6+2*INDEX)
     4        +GENVEL(5)*GENVEL(5+2*INDEX)*(XI1(M)+XI3(M)-XI2(M))
      FI(5) = FI(5)+(XI1(M) -XI3(M) +CGMASS(M)*(XBARCM(3,M)**2
     1        -XBARCM(1,M)**2) )*GENVEL(1)*GENVEL(4)+CGMASS(M)
     2        *XBARCM(2,M)*XBARCM(3,M)*GENVEL(4)*GENVEL(5)-2.0*GENVEL(1)
     3        *CGMASS(M)*XBARCM(3,M)*GENVEL(6+2*INDEX)-2.0*GENVEL(4)
     4        *CGMASS(M)*XBARCM(1,M)*GENVEL(6+2*INDEX)-GENVEL(4)
     5        *GENVEL(5+2*INDEX)*(XI2(M)+XI3(M)-XI1(M))
      FI(6) = FI(6) +GENVEL(4)*(GENVEL(3)*CGMASS(M)+2.0*CGMASS(M)
     1        *GENVEL(6+2*INDEX))-GENVEL(5)*GENVEL(2)*CGMASS(M)
```

```
      FI(5+2*INDEX)=FIPSI(M,0.0,0.0)
      FI(6+2*INDEX)=FIYJ(CGMASS(M),XBARCM(1,M),XBARCM(2,M),
     1                   XBARCM(3,M),0.0)
  100 M=M+1
  200 KDOF = 7 + 2*KAXLES
      INDEXY=KDOF+2
C
      RETURN
      END
```

FORCEI

```
      SUBROUTINE FSPRNG(FWIRE)
C
C     COMPUTE SPRING FORCES.
C
C     OUTPUT -
C     FWIRE  = SPRING GENERALIZED FORCES.
C
      COMMON/CALC/    BETA (3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/CHECK/   JDEBUG
      COMMON/DATAIN/ CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1    DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MOIMDP,NPDAMP(50),SLOPE,
     2    XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DELTAS/ DISP(100)
      COMMON/DIMENS/ MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/FSPDIM/ FORC(100),V(100)
      COMMON XLAMDA(100,44)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IOD(40)
      COMMON/SPRING/ FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1    X(600)
      DIMENSION FWIRE(1)
C
C
C     COMPUTE SPRING FORCES.
C
C     TRANSFORM DISPLACEMENTS FROM GENERALIZED COORDINATES
C            TO SPRING DEFLECTIONS.
C
      CALL MULT (XLAMDA,GENVEL(7),V,   MAXSP,MAXDOF-6,MSPRNG,MDOF-6,1)
      CALL MULT (XLAMDA,GENDIS(7),DISP,MAXSP,MAXDOF-6,MSPRNG,MDOF-6,1)
      IF (JDEBUG.EQ.2.AND.NOUT.EQ.0  ) WRITE (6,888) (GENDIS(IO),
     1  IO=1,MDOF)
      IF (JDEBUG.EQ.2.AND.NOUT.EQ.0) WRITE (6,888) (DISP(IO),IO=1,
     1  MSPRNG)
  888 FORMAT (1X,10E13.6)
C
C     INTERPLOATE IN NON-LINEAR FORCE-DISPLACEMENT TABLES.
C
      DO 50 INDEX=1,6
   50 FWIRE(INDEX)=0.0
C
C     INDEX = SPRING NUMBER.
C
      INDEX=1
C
C     LOOP OVER NON-RACK SPRINGS.
C
C     AXLE SPRINGS
C
      N=0
      DO 90 INDEX1=1,KAXLES
   90 N=N+NASPR(INDEX1)
      DO 100 INDEX1=1,N
      ISTART=JCURVE(INDEX1)
      NP=NPPC(INDEX1)
      CALL INT1(DISP(INDEX),NP,X(ISTART),FOFX(ISTART),FORC (INDEX))
      NPV=NPDAMP(INDEX1)
      ISD=ISDAMP(INDEX1)
      CALL INT1(V(INDEX),NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
      FORC (INDEX)=FORC (INDEX)+FDAMP
      INDEX=INDEX+1
      CALL INT1(DISP(INDEX),NP,X(ISTART),FOFX(ISTART),FORC (INDEX))
      CALL INT1(V(INDEX),NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
      FORC (INDEX)=FORC (INDEX)+FDAMP
      INDEX=INDEX+1
  100 CONTINUE
C
C     BOGIE SPRINGS.
C
```

```
      IF(KBOG.EQ.0) GO TO 120
      DO 110 INDEX1=1,KBOG
C
C     AVERAGE VELOCITY
C
      VDAMP=0.5*(V(INDEX)+V(INDEX+2))
      ISD=ISDAMP(INDEX1+N)
      NPV=NPDAMP(INDEX1+N)
      CALL INT1(VDAMP,NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
C
C     AVERAGE DEFLECTION.
C
      DEFLEC=0.5*(DISP(INDEX)+DISP(INDEX+2))
      ISTART=JCURVE(INDEX1+N)
      NP=NPPC(INDEX1+N)
      CALL INT1(DEFLEC,NP,X(ISTART),FOFX(ISTART),F)
      FORC (INDEX)=0.5*(F+FDAMP)
      FORC (INDEX+2)=FORC (INDEX)
C
C     RIGHT HAND SIDE
C
      INDEX=INDEX+1
      VDAMP=0.5*(V(INDEX)+V(INDEX+2))
      CALL INT1(VDAMP,NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
      DEFLEC=0.5*(DISP(INDEX)+DISP(INDEX+2))
      CALL INT1(DEFLEC,NP,X(ISTART),FOFX(ISTART),F)
      FORC (INDEX)=0.5*(F+FDAMP)
      FORC (INDEX+2)=FORC (INDEX)
  110 INDEX=INDEX+3
C
C     INDEX NOW POINTS TO NEXT SPRING IN TABLE.
C     SET UP SHELTER AND GUY WIRES.
C
C
C     SHELTER SPRINGS.
C
  120 IF(KSHELT+KWIRES.EQ.0) GO TO 150
      K=KSHELT+KWIRES
      INDEX1=2*KBOG+1
      DO 125 INDEX2=1,KAXLES
  125 INDEX1=INDEX1+NASPR(INDEX2)
      DO 140 J=1,K
      ISTART=JCURVE(INDEX1)
      NP=NPPC(INDEX1)
      CALL INT1(DISP(INDEX),NP,X(ISTART),FOFX(ISTART),FORC (INDEX))
      NPV=NPDAMP(INDEX1)
      ISD=ISDAMP(INDEX1)
      CALL INT1(V(INDEX), NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
      FORC (INDEX)=FORC (INDEX)+FDAMP
C
C     RIGHT HAND SIDE.
C
      INDEX=INDEX+1
      CALL INT1(DISP(INDEX),NP,X(ISTART),FOFX(ISTART),FORC (INDEX))
      CALL INT1(V(INDEX),NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
      FORC (INDEX)=FORC (INDEX)+FDAMP
       INDEX=INDEX+1
  140 INDEX1=INDEX1+1
C
C     N = LAST LEFT SIDE SPRING NUMBER.
C
  150 N=INDEX/2-KBOG
C
C     SET UP RACKS - LEFT SIDE FIRST.
C
      IF (KRACKS.EQ.0) GO TO 400
      N=N+1
      DO 200 INDEX1=N,NSPRNG
      ISTART=JCURVE(INDEX1)
      NP=NPPC(INDEX1)
      CALL INT1(DISP(INDEX),NP,X(ISTART),FOFX(ISTART),FORC (INDEX))
      NPV=NPDAMP(INDEX1)
      ISD=ISDAMP(INDEX1)
```

```
      CALL INT1(V(INDEX),NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
      FORC (INDEX)=FORC (INDEX)+FDAMP
  200 INDEX=INDEX+1
C
C     RIGHT HAND SIDE
C
      DO 300 INDEX1=N,NSPRNG
      ISTART=JCURVE(INDEX1)
      NP=NPPC(INDEX1)
      CALL INT1(DISP(INDEX),NP,X(ISTART),FOFX(ISTART),FORC (INDEX))
      NPV=NPDAMP(INDEX1)
      ISD=ISDAMP(INDEX1)
      CALL INT1(V(INDEX),NPV,DAMPV(ISD),DAMPF(ISD),FDAMP)
      FORC (INDEX)=FORC (INDEX)+FDAMP
  300 INDEX=INDEX+1
C
C     TRNSFORM BACK TO GENERALIZED COORDINATES.
C
  400 CALL MULT (FORC ,XLAMDA,FWIRE(7),1,MAXSP,1,MSPRNG,MDOF-6)
      IF (JDEBUG.EQ.2.AND.NOUT.EQ.0) WRITE (6,888) (FORC (IO),IO=1,
     1   MSPRNG)
      IF (JDEBUG.EQ.2.AND.NOUT.EQ.0) WRITE(6,888) (FWIRE(IO),IO=1,MDOF)
C
C     JOB COMPLETE -
C
      RETURN
C
      END
```

FSPRNG

```
      SUBROUTINE GEOM
C
C     SUBROUTINE GEOM READS SHELTER-TRUCK GEOMETRY
C     DATA AND INITIAL CONDITIONS.
C
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/CG/      XCG,YCG,ZCG
      COMMON/CHECK/   JDEBUG
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DIMENS/  MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IOC(40)
      COMMON/PRELOD/  PRELOD(4)
      COMMON /TIREC/  CN,CT,NTIRES(6),NTSPRN,RW,TLX(3,6),TLY(6),TLZ(6),U
      COMMON/WEIGHT/  WEIGHT,WGTS(13)
C
C
      READ (5,14) IOPT
      IF (IOPT.NE.1) WRITE(6,20)
      IF (IOPT.EQ.1) WRITE (6,21)
C
      READ(5,14) KSHELT,KWIRES,KRACKS,KAXLES,KBOG,KRIGID
C
C     KSHELT = NUMBER OF SHELTER ATTACH SPRINGS.
C              (0 = RIGIDLY MOUNTED.)
C
C     KWIRES = NUMBER OF GUY WIRES (MUST BE 0 IF KSHELT = 0).
C
C     KRACKS = NUMBER OF RACKS ON LEFT (PORT) SIDE OF SHELTER.
C              (0 = RIGIDLY MOUNTED.)
C
C     KAXLES = NUMBER OF AXLES ON VEHICLE.
C
C     KBOG   = NUMBER OF BOGIE SPRINGS  ON LEFT SIDE OF TRUCK.
C
C     KRIGID = RIGID BODY FLAG.
C              0 = REGULAR RESPONSE RUN
C              1 = RIGID TRUCK RESPONSE, 3 DOF.
      IF (KRIGID.EQ.1) WRITE (6,38)
C
C     NTIRES = NUMBER OF TIRES ON ONE END OF EACH AXLE.
C     NASPR  = NUMBER OF SPRINGS ATTACHING ONE SIDE OF AXLE TO CHASSIS.
C
      READ (5,14) (NTIRES(IO),IO=1,KAXLES)
      READ(5,14) (NASPR(IO),IO=1,KAXLES)
      IF (KRACKS.GT.MRDIM) GO TO 2400
      IF (KRACKS.GT.0) READ (5,14) (NRSPR(IO),IO=1,KRACKS)
C
C     NRSPR  = NUMBER OF SPRINGS ON EACH RACK.
C
      IF (KSHELT.GT.0) WRITE (6,31) KSHELT
      IF (KSHELT.EQ.0) WRITE (6,27)
      IF (KWIRES.GT.0) WRITE(6,32) KWIRES
      IF (KRACKS.EQ.0) WRITE(6,33)
      IF (KRACKS.GT.0) WRITE (6,34) KRACKS
      IF (KRACKS.GT.0) WRITE (6,35) (IO,NRSPR(IO),IO=1,KRACKS)
      WRITE (6,36) KAXLES
      WRITE(6,5) (IO,NTIRES(IO),NASPR(IO),IO=1,KAXLES)
C
      MDOF=6+2*KAXLES+12*KRACKS
      IF (KSHELT.GT.0) MDOF=MDOF+4
      MSPRNG=2*(KSHELT+KWIRES+KBOG)
      DO 113 INDEX=1,KAXLES
  113 MSPRNG=MSPRNG+2*NASPR(INDEX)
      IF(KRIGID.EQ.1) MDOF=3
      IF(KRIGID.EQ.1) MSPRNG=0
      IF (KRACKS.EQ.0) GO TO 117
```

```
      DO 115 INDEX=1,KRACKS
  115 MSPRNG=MSPRNG+2*NRSPR(INDEX)
C
C     SAVE INDEX FOR TIRE-SPRING DATA.
C
  117 NTSPRN=MSPRNG/2+2
      NSPRNG=MSPRNG/2
      MSPRNG=MSPRNG+2*KBOG
C
C     STRUCTURE.
C
C     MASS DATA.
C
      MASSES=1+KAXLES+KRACKS
      IF (KSHELT.GT.0) MASSES=MASSES+1
      IF (KRIGID.EQ.1) MASSES=1
C
C     DIMENSION CHECKS - SET UP ON VARIABLE
C
C     DIMENSIONS, *******
C
      IF (MDOF.GT.MAXDOF) GO TO 2100
      IF (MSPRNG.GT.MAXSP) GO TO 2200
      IF (MASSES.GT.MAXMAS) GO TO 2300
C
C     STRUCTURE MASS AND POSITION.
C
      DO 120 INDEX=1,MASSES
  120 READ(5,4) CGMASS(INDEX),CGPOS(1,INDEX),CGPOS(2,INDEX),
     1          CGPOS(3,INDEX)
C
C     C.G. CALCULATIONS.
C
      CALL CGCALC
C
C     XBARCM=POSITION OF MASS W.R.T. C.G.
C
      DO 125 INDEX=1,MASSES
      XBARCM(1,INDEX)=CGPOS(1,INDEX)-XCG
      XBARCM(2,INDEX)=CGPOS(2,INDEX)-YCG
      XBARCM(3,INDEX)=CGPOS(3,INDEX)-ZCG
  125 WRITE (6,3) INDEX,(XBARCM(IO,INDEX),IO=1,3)
C
C     INERTIA DATA.
C
      READ (5,4) XIYZ
      WRITE(6,11) XIYZ
      DO 130 IK =1,MASSES
           READ(5,4) XI1(IK),XI2(IK),XI3(IK)
           WRITE(6,16) XI1(IK),XI2(IK),XI3(IK)
  130 CONTINUE
C
C     TAKE CARE OF RHS RACKS.
C
      IF (KRACKS.EQ.0) GO TO 134
      INDEX=2+KAXLES
      IF (KSHELT.GT.0) INDEX=INDEX+1
      J=INDEX+KRACKS
      DO 132 INDEX1=1,KRACKS
      XBARCM(1,J)=CGPOS(1,J)-XCG
      XBARCM(2,J)=CGPOS(2,J)-YCG
      XBARCM(3,J)=CGPOS(3,J)-ZCG
      XI1(J)=XI1(INDEX)
      XI2(J)=XI2(INDEX)
      XI3(J)=XI3(INDEX)
      J=J+1
  132 INDEX=INDEX+1
  134 CONTINUE
C
C     COMPUTE LAMBDA MATRIX.
C
      CALL LAMBDA
      IF (KERR.GT.0) RETURN
```

```
C
C     DAMPING AND STIFFNESS.
C
      IF (KAXLES.EQ.1) GO TO 185
      IF (KWIRES.EQ.0) GO TO 140
      KG = KGWIR2 -KGWIR1 +1
      READ (5,4) (PRELOD(IO),IO=1,KG)
      WRITE(6,6) (PRELOD(IO),IO=1,KG)
  140 CONTINUE
C
C     NON-LINEAR DAMPING CURVES.
C
      IF (KRIGID.EQ.1) GO TO 180
      INDEX1=1
      WRITE (6,9)
      DO 150 INDEX=1,NSPRNG
      READ(5,14) NPDAMP(INDEX)
      ISDAMP(INDEX)=INDEX1
      INDEX2=INDEX1+NPDAMP(INDEX)-1
      IF(MDIMDP.LT.INDEX2) GO TO 2500
      READ(5,4) (DAMPV(IO),DAMPF(IO),IO=INDEX1,INDEX2)
      WRITE (6,7) INDEX,(IO,DAMPV(IO),DAMPF(IO),IO=INDEX1,INDEX2)
  150 INDEX1=INDEX2+1
C     END OF DAMPING LOOP
C
  180 CONTINUE
C
C     TIRE DAMPING COEFFICIENTS.
C
  185 READ (5,4) CN,CT
      WRITE (6,8) CN,CT
C
C     READ NON-LINEAR SPRING FORCE-DISPLACEMENT CURVES.
C     SET SLOPES FOR TRIM PROCEDURE.
C
      CALL INPSPR
      IF (KERR.GT.0) RETURN
C
C     GROUND SLOPE AND GRAVITY, FRONT WHEEL/SURFACE CONTACT POINT.
C
  190 READ(5,4) RW,U,G
      WRITE(6,10) RW,U,G
      READ(5,4) GS(1),GS(2),SLOPE
      WRITE(6,37) GS(1),GS(2),SLOPE
C
C     SET UP WEIGHT TABLE FOR USE IN GRAVITY CALCULATIONS.
C
      WEIGHT=0.0
      MASSES=MASSES+KRACKS
      DO 210 INDEX=1,MASSES
      WGTS(INDEX)=CGMASS(INDEX)*G
  210 WEIGHT=WEIGHT+WGTS(INDEX)
C
C     TIME LIMIT, CYCLE TIME,PRINT FREQUENCY.
C
      READ  (5,4)   ENDTIM,DELTIM,FLAG
      JDEBUG=FLAG
      WRITE (6,12) ENDTIM,DELTIM
      READ(5,14) NPRINT,JGRAPH
      IF(NPRINT.EQ.0) WRITE(6,39)
      IF(NPRINT.GT.0) WRITE(6,40) NPRINT
C
      DELTX = DELTIM
      ENDTX = 0.03
      DELTIM = AMIN1(0.00015,DELTX)
      DELTX1 = DELTIM
      WRITE (6,41) DELTIM,ENDTX
      ENDTX = ENDTX - 0.5*DELTIM
      ENDTIM = ENDTIM - 0.5*DELTIM
      DPRT1 = FLOAT(NPRINT)*DELTX
C
      RETURN
C
```

```
    3 FORMAT (8H INDEX =,I3,25H VECTOR FROM CG TO MASS =,3E20.6)
    4 FORMAT (6F12.1)
    5 FORMAT (22H0AXLE    TIRES   SPRINGS,/,(I5,I6,I8))
    6 FORMAT( 23H0PRELOAD IN GUY WIRES =,2X,5E13.5,/,(1X,10E13.5))
    7 FORMAT ( 7H0SPRINGI3,12X,1HV,13X,1HF/
   1   (12X,I3,1X,2E14.6))
    8 FORMAT (26H0TIRE DAMPING COEFFICIENTS/13H   NORMAL    E15.6/
   1 13H    TANGENTIALE15.6)
    9 FORMAT (//40X,28H DATA FOR NON-LINEAR DAMPING)
   10 FORMAT (18H0RADIUS OF WHEEL =,E13.5,5X,
   1         34H COEFFICIENT OF SLIDING FRICTION =,E13.5,5X,
   2         10H GRAVITY =,E13.5)
   11 FORMAT (14H0INERTIA DATA ,//,9H   XIYZ = ,E13.5,//
   1          ,9H INERTIA ,1X,3HXI1,10X,3HXI2,10X,3HXI3)
   12 FORMAT (24H0PROGRAM WILL RUN UNTIL ,E14.5,
   1           40H SECONDS OF RESPONSE HAVE BEEN COMPUTED.,/,
   2           35H THE TIME INTERVAL BETWEEN STEPS = ,E14.5,9H SECONDS.)
   14 FORMAT (6I12)
   16 FORMAT (4X,10E13.5)
   20 FORMAT (24H0VEHICLE TYPE - WHEELED.)
   21 FORMAT (24H0VEHICLE TYPE - TRACKED.)
   27 FORMAT (25H SHELTER RIGIDLY MOUNTED.)
   31 FORMAT (19H SHELTER MOUNTED ON,I3,9H SPRINGS.)
   32 FORMAT (22H NUMBER OF GUY WIRES =,I3)
   33 FORMAT (23H RACKS RIGIDLY MOUNTED.)
   34 FORMAT (18H0NUMBER OF RACKS =,I3)
   35 FORMAT (12H RACK NUMBER,I3,4H HAS,I3,9H SPRINGS.)
   36 FORMAT (10H0TRUCK HAS,I2,7H AXLES.)
   37 FORMAT(27H GROUND SLOPES ARE  GS(1) =,E14.5,5X,8H GS(2) =,E14.5
   1         ,5X,35HTHE POSITION OF THE SLOPE CHANGE IS ,E14.5)
   38 FORMAT (23H0RIGID RESPONSE, 3 DOF.)
   39 FORMAT (52H0NO TIME HISTORY PRINTOUT. SUMMARY INFORMATION ONLY.)
   40 FORMAT (34H PRINT TIME HISTORY RESPONSE EVERY,I5,7H STEPS.)
   41 FORMAT (28H0A SPECIAL TIME INTERVAL OF E12.5,23H SECONDS IS USED U
     1NTIL E12.5,9H SECONDS.)
C
 2100 WRITE(6,2109)
 2109 FORMAT(33H DIMENSION OVERFLOW - DOF. (GEOM))
      KERR = 3
      RETURN
 2200 WRITE(6,2209)
 2209 FORMAT(45H NUMBER OF SPRINGS OVERFLOW - IN GEOM. (GEOM))
      KERR = 3
      RETURN
 2300 WRITE(6,2309)
 2309 FORMAT(22H MASS OVERFLOW. (GEOM))
      KERR = 3
      RETURN
 2400 WRITE(6,2409)
 2409 FORMAT(32H RACK DIMENSION OVERFLOW. (GEOM))
      KERR = 3
      RETURN
 2500 WRITE(6,2509)
 2509 FORMAT (42H0DIMENSION OVERFLOW - DAMPING DATA. (GEOM))
      KERR = 3
      RETURN
C
      END
```

GEOM

```
      SUBROUTINE GRAV(BETA,FGRAV)
C
C
C     SUBROUTINE GRAV CALCULATES GENERALIZED GRAVITATIONAL FORCES.
C
C     *****      *****
C
C
C     INPUT -
C     BETA = TRANSFORMATION MATRIX.
C
C     OUTPUT -
C     FGRAV  = GRAVITATIONAL GENERALIZED FORCE.
C
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
      COMMON/WEIGHT/ WEIGHT,WGTS(13)
C
      DIMENSION BETA(3,3),FGRAV(1)
C
      FGRAV(1)  =0.0
      FGRAV(2)  =WEIGHT*BETA(2,1)
      FGRAV(3)  =WEIGHT*BETA(2,2)
      IF (MASSES.EQ.1) RETURN
      FGRAV(4)  =0.0
      FGRAV(5)  =0.0
      FGRAV(6)=WEIGHT*BETA(2,3)
      FGRAV(7)=0.0
      FGRAV(8)=WGTS(2)*BETA(2,2)
      FGRAV(9)=0.0
      FGRAV(10)=WGTS(3)*BETA(2,2)
      M=4
      INDEX=11
C
C     AXLE LOOP
C
      IF (KAXLES.LE.2) GO TO 100
      DO 50 INDEX1=3,KAXLES
      FGRAV(INDEX)=0.0
      FGRAV(INDEX+1)=WGTS(M)*BETA(2,2)
      INDEX=INDEX+2
   50 M=M+1
C
  100 IF (KSHELT.EQ.0) GO TO 200
      FGRAV(INDEX)=0.0
      FGRAV(INDEX+1)=WGTS(M)*BETA(2,2)
      FGRAV(INDEX+2)=0.0
      FGRAV(INDEX+3)=WGTS(M)*BETA(2,3)
      INDEX=INDEX+4
      M=M+1
C
  200 IF (KRACKS.EQ.0) RETURN
      DO 300 INDEX1=M,MASSES
      FGRAV(INDEX)=0.0
      FGRAV(INDEX+1)=WGTS(INDEX1)*BETA(2,1)
      FGRAV(INDEX+2)=WGTS(INDEX1)*BETA(2,2)
      FGRAV(INDEX+3)=0.0
      FGRAV(INDEX+4)=0.0
      FGRAV(INDEX+5)=WGTS(INDEX1)*BETA(2,3)
  300 INDEX=INDEX+6
C
      RETURN
C
      END
```

GRAV

```
      SUBROUTINE HYTRUK (T,R,P,Q,IOPT)
C     BLAST DATA FOR TRUCK PROGRAM.
C     KBLAST - 1, 1 KT AT SEA LEVEL, BASED ON BRL REPORT NO. 1889 BY
C                 N. ETHRIDGE, P. 22.
C            - 2, 60 M W1/3, BY N. ETHRIDGE, OCT., 1976.
C
C     IOPT 1 - GIVEN SHOCK RADIUS (R), RETURN PEAK PRESSURE (P) AND
C                PEAK DYNAMIC PRESSURE (Q)
C     IOPT 2 - MUST HAVE CALLED IOPT=1 OR 3 FIRST.  GIVEN TIME, RETURNS
C                PRESSURE (P) AND DYNAMIC PRESSURE (Q)
C     IOPT 3 - GIVEN PEAK OVERPRESSURE (P), RETURN SHOCK RADIUS (R)
C                AND PEAK DYNAMIC PRESSURE (Q)
C     FOR OPTIONS 1 AND 3, THE OVERPRESSURE IMPULSE (PIMP), AND THE
C       DYNAMIC PRESSURE IMPULSE (QIMP) ARE ALSO CALCULATED.
C       ALSO, THE POSITIVE PHASE DURATION (T).
C     FORMAL PARAMETER UNITS - R,FT  P,PSI, Q,PSI.
C     PIMP, PSI-SEC, QIMP,PSI-SEC.
C     LOG-LOG INTERPOLATION.
C     ON FIRST PASS ALL TABLES ARE CONVERTED TO LOGS.
C     PRESSURE TABLES IN PSI, RADIUS TABLE IN FT.  TIMES IN SECONDS.
C       IF OTHER UNITS ARE USED, C1 AND C2 MUST BE CHANGED ACCORDINGLY.
C     CONVERSIONS - C1(LENGTH), C2(PRESSURE)
      COMMON/BLAST/  A,AD,AD1,AD2,ALT,A0,COSA,DELTA,KBLAST,PIMP,PR,
     1   PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      DIMENSION RT(27),PST(27),CPST(27),CPQT(27),TDT(27)
      DIMENSION RT1(22),PST1(22),CPST1(22),CPQT1(22),TDT1(22)
      DIMENSION RT2(27),PST2(27),CPST2(27),CPQT2(27),TDT2(27)
C
      DATA C1/1.0/, C2/1.0/
      DATA NT1/22/, NT2/27/
      DATA RT1/ 3310.2,2675.0, 2270.1,2000.1,1800.0,1650.0,1545.0,
     1  1445.0,1370.1,1305.2,1190.0,1110.0,982.0,897.0,799.9,700.2,
     2  630.0,580.1,542.0,509.9,484.9,462.0/
      DATA PST1/1.5,2.0,2.5,3.0,3.5,4.0,4.5,5.0,5.5,6.0,7.0,8.0,
     1  10.0,12.0,15.0,20.0,25.0,30.0,35.0,40.0,45.0,50.0/
      DATA CPST1/3.1,3.8,4.47,5.05,5.7,6.25,6.8,7.3,7.78,8.22,
     1  9.1,9.85,11.4,12.8,15.,18.4,21.75,25.,28.3,31.5,
     2  34.9,38.2/
      DATA CPQT1/5.5,6.62,7.7,8.7,9.7,10.65,11.5,12.4,13.2,
     1  13.9,15.25,16.45,18.6,20.75,23.7,28.4,32.9,37.3,41.9,
     2  46.5,51.,55.7/
      DATA TDT1/.407,.379,.358,.340,.329,.316,.307,.299,.291,
     1  .285,.273,.264,.248,.237,.226,.206,.186,.166,.155,
     2  .151,.150,.152/
C
      DATA RT2/2247.38,2043.96,1886.48,1761.81,1650.26,1564.96,
     1  1486.22,1420.6,1361.55,1312.34,1259.84,1177.82,1105.64,
     2  1043.29,990.81,951.43,908.78,874.34,844.81,798.67,746.38,707.01,
     3  659.44,600.39,554.46,521.65,467.8/
      DATA PST2/3.,3.5,4.,4.5,5.,5.5,6.,6.5,7.,7.5,8.,9.,10.,11.,12.,
     1  13.,14.,15.,16.,18.,20.,22.,25.,30.,35.,40.,50./
      DATA CPST2/4.02,4.56,4.96,5.39,5.28,6.22,6.54,6.96,7.26,7.62,7.88,
     1  8.5,9.04,9.5,10.,10.55,11.,11.46,12.03,12.76,13.74,14.62,15.71,
     2  17.56,19.16,20.85,23.82/
      DATA CPQT2/8.54,9.24,9.86,10.53,10.93,11.51,12.,12.47,12.85,
     1  13.36,13.61,14.21,14.97,15.54,16.18,17.14,17.54,18.25,18.97,
     2  20.14,21.34,22.39,24.02,26.9,29.53,32.66,38./
      DATA TDT2/.383,.365,.349,.336,.324,.315,.303,.295,.287,.279,
     1  .271,.259,.247,.236,.227,.219,.211,.205,.199,.186,.179,
     2  .171,.160,.147,.135,.127,.113/
      DATA KBL/-1/
C
      IF (KBLAST.EQ.KBL) GO TO 100
      KBL = KBLAST
      GO TO (20,60), KBLAST
   20 NT = NT1
      DO 50 I=1,NT
      RT(I) = ALOG(RT1(I))
      PST(I) = ALOG(PST1(I))
      CPST(I) = ALOG(CPST1(I))
      CPQT(I) = ALOG(CPQT1(I))
   50 TDT(I) = ALOG(TDT1(I))
      GO TO 100
```

```
C
   60 NT = NT2
      DO 70 I=1,NT
      RT(I) = ALOG(RT2(I))
      PST(I) = ALOG(PST2(I))
      CPST(I) = ALOG(CPST2(I))
      CPQT(I) = ALOG(CPQT2(I))
   70 TDT(I) = ALOG(TDT2(I))
C
  100 GO TO (200,300,700), IOPT
C
C     OPTION 1.
C
  200 RR = R/(C1*SF(1))
      RL = ALOG(RR)
      IF (RL.LT.RT(NT).OR.RL.GT.RT(1)) GO TO 800
  220 CALL INT1X (RL,NT,RT,J,FR,2)
      PP = EXP(PST(J) + FR*(PST(J+1) - PST(J)))
      P = PP*SF(3)*C2
C
  250 CPS = EXP(CPST(J) + FR*(CPST(J+1) - CPST(J)))
      X = PP*C2/14.696
      PQ = (36.74/C2)*X**2/(7.0+X)
      CPQ = EXP(CPQT(J) + FR*(CPQT(J+1) - CPQT(J)))
      TD = EXP(TDT(J) + FR*(TDT(J+1) - TDT(J)))
      TS = TD - 1.0/CPS
      TSQ = TD - 1.0/CPQ
      PS1 = PP*EXP(-CPS*TS)*CPS
      PQ1 = PQ*EXP(-CPQ*TSQ)*CPQ
      Q = PQ*SF(3)*C2
      PIMP = PP*(1.0-0.5*EXP(-CPS*TS))/CPS
      QIMP = PQ*(1.0 - 0.5*EXP(-CPQ*TSQ))/CPQ
      PIMP = PIMP*C2*SF(2)*SF(3)
      QIMP = QIMP*C2*SF(2)*SF(3)
      T = TD*SF(2)
      RETURN
C
C     OPTION 2.
C
  300 TT = T/SF(2)
      IF (TT.GT.TSQ) GO TO 400
      Q = PQ*EXP(-CPQ*TT)
      IF (TT.GT.TS) GC TO 450
      P = PP*EXP(-CPS*TT)
      GO TO 600
  400 IF (TT.GT.TD) GO TO 500
      Q = PQ1*(TD - TT)
  450 P = PS1*(TD - TT)
      GO TO 600
  500 P = 0.
      Q = 0.
  600 P = P*SF(3)*C2
      Q = Q*SF(3)*C2
      RETURN
C
C     OPTION 3.
C
  700 PP = P/(SF(3)*C2)
      PL = ALOG(PP)
      IF (PL.LT.PST(1).OR.PL.GT.PST(NT)) GO TO 900
  750 CALL INT1X (PL,NT,PST,J,FR,1)
      R = EXP(RT(J) + FR*(RT(J+1) - RT(J)))
      R = R*SF(1)*C1
      GO TO 250
C
C
C     BEYOND TABLE LIMITS.
C
  800 WRITE (6,2000) R,RL,RT(1),RT(NT)
      IF (RL.LT.RT(NT)) RL = RT(NT)
      IF (RL.GT.RT(1)) RL = RT(1)
      RR = EXP(RL)
      R = RR*C1*SF(1)
```

```
      GO TO 220
  900 WRITE (6,2100) P,PL,PST(1),PST(NT)
      IF (PL.LT.PST(1)) PL = PST(1)
      IF (PL.GT.PST(NT)) PL = PST(NT)
      PP = EXP(PL)
      P = PP*SF(3)*C2
      GO TO 750
C
 2000 FORMAT (30HORANGE OUT OF BOUNDS IN HYTRUK 4E15.6/
     1   20H VALUE AT LIMIT USED)
 2100 FORMAT (33HOPRESSURE OUT OF BOUNDS IN HYTRUK 4E15.6/
     1   20H VALUE AT LIMIT USED)
      END
```

HYTRUK

```
      SUBROUTINE INLOAD
C     INPUT DATA FOR BLAST LOADING ON TRUCK-SHELTER.
      COMMON/BLAST/  A,AD,AD1,AD2,ALT,A0,COSA,DELTA,KBLAST,PIMP,PR,
     1   PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/CG/     XCG,YCG,ZCG
      COMMON/LOAD/    AL(6),A6(6),CK(6),CP(6),DDCODE(6),
     1   EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2   NPS(6,4),PI,PSINF(6),PS00(6),S(16,6,4),SIG(6),
     3   SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,6,4),
     4   SJ1(16,6,4),SJ2(16,6,4),SJ3(16,6,4),SJ4(16,6,4),
     5   TD(16,6,4),WG(6),XBOX(16,6,4),YBOX(16,6,4),
     6   ZBOX(16,6,4), TDLAST
C
C     NUMBER OF BOX CONFIGURATIONS.
      READ (5,2) NBOX,KDEBUG
C     TRUCK OR SHELTER CODE FOR EACH BOX.
      READ (5,2) (KTS(N),N=1,NBOX)
      WRITE (6,1050) NBOX
      WRITE (6,1200)
      DO 30 N=1,NBOX
      KT = KTS(N)
      IF (KT.EQ.1) WRITE (6,1250) N,KT
      IF (KT.EQ.2) WRITE (6,1260) N,KT
   30 CONTINUE
      WRITE (6,1300)
      DO 200 N=1,NBOX
      DO 200 I=1,6
      IF (I.NE.2) GO TO 80
C     FAR SIDE.
      NPSS = NPS(1,N)
      NPS(2,N) = NPSS
      IF (NPSS.EQ.0) GO TO 200
      DO 50 K=1,NPSS
      XBOX(K,2,N) = -XBOX(K,1,N)
      YBOX(K,2,N) = YBOX(K,1,N)
      ZBOX(K,2,N) = ZBOX(K,1,N)
      S(K,2,N) = S(K,1,N)
      SI1(K,2,N) = SI1(K,1,N)
      SI2(K,2,N) = SI2(K,1,N)
      SI3(K,2,N) = SI3(K,1,N)
   50 SI4(K,2,N) = SI4(K,1,N)
      GO TO 200
   80 CONTINUE
      READ(5,2) NPS(I,N)
      NPSS = NPS(I,N)
      IF (NPSS.EQ.0) GO TO 200
      DO 100 K=1,NPSS
      READ (5,1) XB,YB,ZB,S(K,I,N)
      READ (5,1) SI1(K,I,N),SI2(K,I,N),SI3(K,I,N),SI4(K,I,N)
      WRITE (6,1400) N,I,K,XB,YB,ZB,S(K,I,N),SI1(K,I,N),SI2(K,I,N),
     1  SI3(K,I,N),SI4(K,I,N)
      XBOX(K,I,N) = XB     - XCG
      YBOX(K,I,N) = YB     - YCG
      ZBOX(K,I,N) = ZB     - ZCG
  100 CONTINUE
  200 CONTINUE
      RETURN
C
    1 FORMAT (6F12.1)
    2 FORMAT (6I12)
 1050 FORMAT (44H0NUMBER OF AERODYNAMIC BOX CONFIGURATIONS = I3)
 1200 FORMAT (21H0BOX DESIGNATION CODE/16H   BOX NO.  CODE)
 1250 FORMAT (1X,I6,6X,I2,3X,31HBOX RIGIDLY ATTACHED TO VEHICLE)
 1260 FORMAT (1X,I6,6X,I2,3X,45HBOX (SHELTER) NOT RIGIDLY ATTACHED TO VE
     1HICLE)
 1300 FORMAT (23H0BOX,FACE,GRID POINT, X,11X,1HY,14X,1HZ,12X,4HAREA,12X,
     1   3HSI1,12X,3HSI2,12X,3HSI3,12X,3HSI4)
 1400 FORMAT (3I3,4E15.6,1X,4E15.6)
      END
```

INLOAD

```
      SUBROUTINE INPSPR
C
C
C     SUBROUTINE INPSPR READS IN SPRING DATA AND SETS UP ARRAYS
C        OF BOTTOMING POSITIONS.
C
      COMMON/BOTM/    EXPAN(100),NBOTOM(100),SBOTM(100),VEL(100)
      COMMON/DIMENS/ MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IOD(40)
      COMMON/SPRING/ FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1   X(600)
      COMMON /TIREC/ CN,CT,NTIRES(6),NTSPRN,RW,TLX(3,6),TLY(6),TLZ(6),U
C
      IPOINT=1
      WRITE (6,17)
C
C     INCLUDE TIRE DATA.
C
      DO 200 INDEX=1,NTSPRN
      READ  (5,14) NP
      NPPC(INDEX)=NP
      IEND=IPOINT+NP-1
      IF (IEND.GT.MAXSPR) GO TO 1000
      WRITE (6,15) INDEX
C
      READ  (5,4)   (X(IO),FOFX(IO),IO=IPOINT,IEND)
      IO=IPOINT
      DO 180 IO1=1,NP
      WRITE (6,23) IO1,X(IO),FOFX(IO)
  180 IO=IO+1
C
      JCURVE(INDEX)=IPOINT
      IPOINT=IEND+1
  200 CONTINUE
C
C     SAVE DATA FOR SPRING BOTTOMING ROUTINE.
C
      N=KSHELT+KWIRES+KBOG
      NA=0
      DO 210 IDUM=1,KAXLES
  210 NA=NA+NASPR(IDUM)
      N=N+NA
      INDEX1=1
      DO 300 INDEX=1,N
      M=JCURVE(INDEX)
      SBOTM(INDEX1)=X(M)
      SBOTM(INDEX1+1)=X(M)
      MM=NPPC(INDEX)+M-1
      EXPAN(INDEX1)=X(MM)
      EXPAN(INDEX1+1)=X(MM)
      IF(KBOG.EQ.0) GO TO 300
      IF(INDEX.LE.NA.OR.INDEX.GT.NA+KBOG) GO TO 300
      INDEX1=INDEX1+2
      SBOTM(INDEX1)=X(M)
      SBOTM(INDEX1+1)=X(M)
      EXPAN(INDEX1)=X(MM)
      EXPAN(INDEX1+1)=X(MM)
  300 INDEX1=INDEX1+2
C
C     RACK DATA.
C
      K=NSPRNG-N
      N=N+1
      DO 400 INDEX=N,NSPRNG
      M=JCURVE(INDEX)
      SBOTM(INDEX1)=X(M)
      SBOTM(INDEX1+K)=X(M)
      MM=NPPC(INDEX)+M-1
      EXPAN(INDEX1)=X(MM)
      EXPAN(INDEX1+K)=X(MM)
```

```
  400 INDEX1=INDEX1+1
C
      RETURN
C
 1000 WRITE (6,18) MAXSPR
      KERR = 3
      RETURN
C
    4 FORMAT (6F12.1)
   14 FORMAT (6I12)
   15 FORMAT (7H0SPRING,I3,11X,1HX,13X,4HF(X))
   17 FORMAT (1H0,40X,28HDATA FOR NON-LINEAR SPRINGS.)
   18 FORMAT (10(2H *),48H NON-LINEAR SPRING DATA TABLE OVERFLOW. (INPSP
     1R)/
     1          10(2H *),20H COMMON BLOCK SPRING,/,
     2          10(2H *),34H MAXSPR = DIMENSION OF FOFK AND X.,/,
     3          10(2H *),20H CURRENT DIMENSION =,I4)
   23 FORMAT (12X,I3,1X,2E14.5)
C
      END
```

INFSPR

```
      SUBROUTINE INT1 (X, NX,XT,YT,R)
C     SUBROUTINE INT1 IS A LINEAR INTERPOLATION ROUTINE.
C     GIVEN A VALUE X, INT1 RETURNS THE CORRESPONDING VALUE Y.
C     NX = DIMENSION OF X-TABLE AND Y-TABLE IN CALLING PROGRAM.
C     XT = TABLE OF X-VALUES IN CALLING PROGRAM.
C     YT = TABLE OF Y-VALUES IN CALLING PROGRAM.
C     R = RETURN VALUE Y.
      COMMON/CINT1/  IIMAX,IINT
      DIMENSION XT(NX),YT(NX)
      DATA C/0.1/
C
      IF (X-XT(1)) 1,9,2
    1 R=YT(2)+((X-XT(2))/(XT(1)-XT(2)))*(YT(1)-YT(2))
      IF (ABS((X-XT(1))/(XT(1)-XT(NX))).LT.C) RETURN
      IINT = IINT + 1
      IF (IINT.EQ.IIMAX+1) WRITE (6,20)
      IF (IINT.GT.IIMAX) RETURN
      WRITE (6,8) X,R,     (XT(I),I=1,NX)
      RETURN
    2 IF (X-XT(NX)) 4,10,3
    3 R=YT(NX-1)+((X-XT(NX-1))/(XT(NX)-XT(NX-1)))*(YT(NX)-YT(NX-1))
      IF (ABS((X-XT(NX))/(XT(1)-XT(NX))).LT.C) RETURN
      IINT = IINT + 1
      IF (IINT.EQ.IIMAX+1) WRITE (6,20)
      IF (IINT.GT.IIMAX) RETURN
      WRITE (6,8) X,R,     (XT(I),I=1,NX)
      RETURN
    4 DO 5  I=2,NX
      IF (X-XT(I)) 7,6,5
    5 CONTINUE
    6 R=YT(I)
      RETURN
    7 R=YT(I-1)+(X-XT(I-1))*(YT(I)-YT(I-1))/(XT(I)-XT(I-1))
      RETURN
    8 FORMAT (9H X-VALUE E14.5/35H OUTSIDE TABLE. Y-VALUE RETURNED = E14
     1.5/10H X-TABLE =/(6E14.5))
    9 R=YT(1)
      RETURN
   10 R=YT(NX)
      RETURN
C
   20 FORMAT (58H0**********WARNING - SUBSEQUENT MESSAGES SUPPRESSED IN I
     1NT1)
      END
```

INT1

```
      SUBROUTINE INT1X (X,NX,XT,J,FR,K)
C     SPECIAL 1 DIMENSIONAL INTERPOLATION ROUTINE.
C     X - INPUT VARIABLE.
C     NX - NUMBER OF TABLE ENTRIES.
C     XT - TABLE OF X
C     J - LOWER INDEX IN TABLE WHICH BRACKETS X.
C     FR - FRACTION OF DISTANCE BETWEEN J AND J+1 ENTRIES.
C     K - CODE
C         1, TABLE IN ASCENDING ORDER.
C         2, TABLE IN DESCENDING ORDER.
C
      DIMENSION XT(1)
      CK = 1.
      IF (K.EQ.2) CK = -1.
      J = 1
      DO 200 I=1,NX
      IF ((X-XT(I))*CK) 300,300,100
  100 J = I
  200 CONTINUE
  300 FR = (X - XT(J))/(XT(J+1) - XT(J))
      RETURN
      END
```

INT1X

```
      SUBROUTINE LAMBDA
C
C
C     SUBROUTINE LAMBDA CALCULATES MATRIX OF SPRING DISPLACEMENTS
C        DUE TO GENERALIZED DISPLACEMENTS.
C
      COMMON/CG/      XCG,YCG,ZCG
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON XLAMDA(100,44)
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
      COMMON /TIREC/ CN,CT,NTIRES(6),NTSPRN,RW,TLX(3,6),TLY(6),TLZ(6),U
C
      IF (KAXLES.EQ.1) GO TO 1000
      NDOF=MDOF-6
      DO 50  INDEX=1,NDOF
      DO 50  INDEX1=1,MSPRNG
   50 XLAMDA(INDEX1,INDEX)=0.0
      IDOF=1
      KGWIR1=0
      KGWIR2=0
C
C     LOOP OVER AXLE SPRINGS.
C
      WRITE (6,6)
      INDEX=1
      IO=1
      DO 75 IDUM=1,KAXLES
      N=NASPR(IDUM)
      IF (N.EQ.0) GO TO 75
      DO 70 IDUM1=1,N
      READ(5,1) RLX,RLY,RLZ,SLX,SLY,SLZ
      WRITE(6,2) IO,RLX,RLY,RLZ,SLX,SLY,SLZ
      IO=IO+1
C
      TL=SQRT((SLX-RLX)**2+(SLY-RLY)**2+(SLZ-RLZ)**2)
      A=(SLX-RLX)/TL
      B=(SLY-RLY)/TL
      SLX=SLX-CGPOS(1,IDUM+1)
      SLY=SLY-CGPOS(2,IDUM+1)
      XLAMDA(INDEX,IDOF)=SLX*B-SLY*A
      XLAMDA(INDEX,IDOF+1)=B
C
C     RIGHT HAND SIDE.
C
      XLAMDA(INDEX+1,IDOF)=-XLAMDA(INDEX,IDOF)
      XLAMDA(INDEX+1,IDOF+1)=XLAMDA(INDEX,IDOF+1)
   70 INDEX=INDEX+2
   75 IDOF=IDOF+2
C
C     SAVE TIRE DATA.
C
      DO 100 IDUM=1,KAXLES
      READ (5,1) TLYB,TLZB
      NTT=NTIRES(IDUM)
      READ(5,1)  (TLX(NT,IDUM),NT=1,NTT)
      DO 80 NT=1,NTT
      WRITE (6,7) TLX(NT,IDUM),TLYB,TLZB
      TLX(NT,IDUM) = TLX(NT,IDUM) - XCG
      IF (NT.GT.1) GO TO 80
      TLY(IDUM) = TLYB-YCG
      TLZ(IDUM) = TLZB - ZCG
   80 WRITE(6,8) TLX(NT,IDUM),TLY(IDUM),TLZ(IDUM)
  100 CONTINUE
C
C     BOGIE SPRINGS.
C
      IF(KBOG.EQ.0) GO TO 120
      WRITE (6,12)
```

```
      DO 112 IDUM=1,KBOG
      READ(5,9) KAXLB(IDUM),XBOGIE(IDUM)
      IF(KAXLB(IDUM).GT.0 .AND. KAXLB(IDUM).LT.KAXLES) GO TO 110
      WRITE(6,10) KAXLB(IDUM),XBOGIE(IDUM)
      KERR = 3
      RETURN
C
  110 WRITE (6,11) IO,KAXLB(IDUM),XBOGIE(IDUM)
  112 IO = IO + 1
C
C     SET UP LAMBDA FOR BOGIE SPRINGS.
C
      DO 115 IDUM=1,KBOG
      J=KAXLB(IDUM)
      IDOF=2*J-1
C     AXLE A.
C
C     TL=1.0, A=0.0, B=1.0
C
      SLXA=XBOGIE(IDUM)
      XLAMDA(INDEX,IDOF)=-SLXA
      XLAMDA(INDEX,IDOF+1)=-1.0
C
C     RIGHT HAND SIDE.
C
      XLAMDA(INDEX+1,IDOF)=SLXA
      XLAMDA(INDEX+1,IDOF+1)=-1.0
      INDEX=INDEX+2
C
C     AXLE B.
C
C     TL=1.0, A=0.0, B=1.0
C
      XLAMDA(INDEX,IDOF+2)=-SLXA
      XLAMDA(INDEX,IDOF+3)=-1.0
C
C     RIGHT HAND SIDE
C
      XLAMDA(INDEX+1,IDOF+2)= SLXA
      XLAMDA(INDEX+1,IDOF+3)=-1.0
  115 INDEX=INDEX+2
C
C     KS1 = FIRST SHELTER DOF, ROLL.
C     KS2 = SECOND SHELTER DOF, HEAVE.
C     KS3 = THIRD SHELTER DOF, PITCH.
C     KS4 = FOURTH SHELTER DOF, FORE AND AFT.
C
  120 CONTINUE
      KS1=2*KAXLES+1
      KS2=KS1+1
      KS3=KS2+1
      KS4=KS3+1
      IDOF=KS1-1
      K=KAXLES+1
C
C     IF SHELTER IS RIGIDLY ATTACHED, THEN NO SHELTER SPRINGS.
C
      IF (KSHELT.EQ.0) GO TO 200
      K=K+1
C
C     LOOP OVER NUMBER OF SHELTER SPRINGS.
C
      WRITE(6,3)
      DO 125 INDEX1=1,KSHELT
      READ(5,1) RLX,RLY,RLZ,SLX,SLY,SLZ
      WRITE(6,2) IO,RLX,RLY,RLZ,SLX,SLY,SLZ
      IO=IO+1
      TL = SQRT((SLX-RLX)**2+(SLY-RLY)**2+(SLZ-RLZ)**2)
      A=(SLX-RLX)/TL
      B=(SLY-RLY)/TL
      C=(SLZ-RLZ)/TL
      SLX=SLX-CGPOS(1,K)
      SLY=SLY-CGPOS(2,K)
```

```
      SLZ=SLZ-CGPOS(3,K)
      XLAMDA(INDEX,KS1) =SLX*B-SLY*A
      XLAMDA(INDEX,KS2)=B
      XLAMDA(INDEX,KS3)=SLY*C-SLZ*B
C
      XLAMDA(INDEX,KS4)=C
C     RIGHT HAND SIDE.
C
      XLAMDA(INDEX+1,KS1)=-XLAMDA(INDEX,KS1)
      XLAMDA(INDEX+1,KS2) = XLAMDA(INDEX,KS2)
      XLAMDA(INDEX+1,KS3) = XLAMDA(INDEX,KS3)
      XLAMDA(INDEX+1,KS4) = XLAMDA(INDEX,KS4)
C
C     INCREMENT SPRING COUNT.
C
  125 INDEX=INDEX+2
      IDOF=KS4
C
C     GUY WIRES.
C
      IF (KWIRES.EQ.0) GO TO 200
      WRITE(6,5)
      KGWIR1 = IO
      KGWIR2=KGWIR1-1+KWIRES
      DO 150 INDEX1=1,KWIRES
      READ(5,1) RLX,RLY,RLZ,SLX,SLY,SLZ
      WRITE(6,2) IO,RLX,RLY,RLZ,SLX,SLY,SLZ
      IO=IO+1
      TL=SQRT((SLX-RLX)**2+(SLY-RLY)**2+(SLZ-RLZ)**2)
      A=(SLX-RLX)/TL
      B=(SLY-RLY)/TL
      C=(SLZ-RLZ)/TL
      SLX=SLX-CGPOS(1,K)
      SLY=SLY-CGPOS(2,K)
      SLZ=SLZ-CGPOS(3,K)
      XLAMDA(INDEX,KS1) =SLX*B-SLY*A
      XLAMDA(INDEX,KS2)=B
      XLAMDA(INDEX,KS3)=SLY*C-SLZ*B
      XLAMDA(INDEX,KS4)=C
      XLAMDA(INDEX+1,KS1)=-XLAMDA(INDEX,KS1)
      XLAMDA(INDEX+1,KS2)=XLAMDA(INDEX,KS2)
      XLAMDA(INDEX+1,KS3)=XLAMDA(INDEX,KS3)
      XLAMDA(INDEX+1,KS4)=XLAMDA(INDEX,KS4)
  150 INDEX=INDEX+2
C
C     RACKS.
C
  200 IF(KRACKS.EQ.0) RETURN
      WRITE(6,4)
C
C     SAVE PORT INDICES FOR STARBOARD RACKS.
C
      IDOFS=IDOF+1
      IDOF=IDOFS
      INDEXS=INDEX
      KS=K
      DO 400 INDEX1=1,KRACKS
      K=K+1
C
C     NUMBER OF SPRINGS FOR EACH RACK.
C
      N=NRSPR(INDEX1)
      DO 300 INDEX2=1,N
      READ(5,1) RLX,RLY,RLZ,SLX,SLY,SLZ
      WRITE(6,2) IO,RLX,RLY,RLZ,SLX,SLY,SLZ
      IO=IO+1
      TL=SQRT((SLX-RLX)**2+(SLY-RLY)**2+(SLZ-RLZ)**2)
      A=(SLX-RLX)/TL
      B=(SLY-RLY)/TL
      C=(SLZ-RLZ)/TL
      RLX=RLX-CGPOS(1,K)
      RLY=RLY-CGPOS(2,K)
      RLZ=RLZ-CGPOS(3,K)
```

```
      XLAMDA(INDEX,IDOF)=RLY*A-RLX*B
      XLAMDA(INDEX,IDOF+1)=-A
      XLAMDA(INDEX,IDOF+2)=-B
      XLAMDA(INDEX,IDOF+3)=RLZ*B-RLY*C
      XLAMDA(INDEX,IDOF+4)=RLX*C-RLZ*A
      XLAMDA(INDEX,IDOF+5)=-C
      IF(KSHELT.EQ.0) GO TO 300
C
C     SHELTER IS NOT RIGIDLY ATTACHED.
C
      SLX=SLX-CGPOS(1,KS)
      SLY=SLY-CGPOS(2,KS)
      SLZ=SLZ-CGPOS(3,KS)
      XLAMDA(INDEX,KS1) =SLX*B-SLY*A
      XLAMDA(INDEX,KS2)=B
      XLAMDA(INDEX,KS3)=SLY*C-SLZ*B
      XLAMDA(INDEX,KS4)=C
C
C     INCREMENT SPRING COUNT.
  300 INDEX=INDEX+1
  400 IDOF=IDOF+6
C
C     TAKE CARE OF STARBOARD RACKS.
C
C     LOOP OVER RACKS.
C
      DO 600 INDEX1=1,KRACKS
      N=NRSPR(INDEX1)
C
C     LOOP OVER SPRINGS.
C
      DO 500 INDEX2=1,N
      XLAMDA(INDEX,IDOF)=-XLAMDA(INDEXS,IDOFS)
      XLAMDA(INDEX,IDOF+1)=-XLAMDA(INDEXS,IDOFS+1)
      XLAMDA(INDEX,IDOF+2)=XLAMDA(INDEXS,IDOFS+2)
      XLAMDA(INDEX,IDOF+3)=XLAMDA(INDEXS,IDOFS+3)
      XLAMDA(INDEX,IDOF+4)=-XLAMDA(INDEXS,IDOFS+4)
      XLAMDA(INDEX,IDOF+5)=XLAMDA(INDEXS,IDOFS+5)
      IF (KSHELT.EQ.0) GO TO 450
      XLAMDA(INDEX,KS1)=-XLAMDA(INDEXS,KS1)
      XLAMDA(INDEX,KS2)=XLAMDA(INDEXS,KS2)
      XLAMDA(INDEX,KS3)=XLAMDA(INDEXS,KS3)
      XLAMDA(INDEX,KS4)=XLAMDA(INDEXS,KS4)
  450 INDEXS=INDEXS+1
  500 INDEX=INDEX+1
      IDOF=IDOF+6
  600 IDOFS=IDOFS+6
C
      RETURN
C
C     RIGID CASE - TIRE DATA
C
 1000 READ (5,1) TLYB,TLZB
      NTT=NTIRES(1)
      READ(5,1) (TLX(NT,1),NT=1,NTT)
      DO 1010 NT=1,NTT
      WRITE (6,7) TLX(NT,1),TLYB,TLZB
      TLX(NT,1) = TLX(NT,1) - XCG
      IF (NT.GT.1) GO TO 1010
      TLY(1) = TLYB-YCG
      TLZ(1) = TLZB-ZCG
 1010 WRITE(6,8) TLX(NT,1),TLY(1),TLZ(1)
      RETURN
C
C     FORMAT STATEMENTS.
C
    1 FORMAT(6F12.1)
    2 FORMAT(1X,I10,6E15.5)
    3 FORMAT(33H0SHELTER SPRING ATTACHMENT POINTS/6X,6HSPRING)
    4 FORMAT(30H0RACK SPRING ATTACHMENT POINTS/6X,6HSPRING)
    5 FORMAT(34H0GUY WIRE SPRING ATTACHMENT POINTS/6X,6HSPRING)
    6 FORMAT (25H0SPRING ATTACHMENT POINTS)
    7 FORMAT (5H0TIRE,  6X,6E15.5)
```

LAMBDA

```
 8 FORMAT (1X,9HTIRE-(CG),1X,6E15.5)
 9 FORMAT (I12,F12.1)
10 FORMAT (1H0,10(2H* ),45HILLEGAL VALUE FOR BOGIE AXLE NUMBER. (LAMB
  1DA),
  1         /,21X,6HAXLE =,I3,11H X ATTACH =,E13.5)
11 FORMAT (8X,I2,6X,I2,E15.6)
12 FORMAT (31H0BOGIE SPRING ATTACHMENT POINTS//6X,6HSPRING,2X,
  1   4HAXLE,9X,1HX)
C
   END
```

LAMBDA

```
      SUBROUTINE MASSM
C
C     SUBROUTINE MASSM CALCULATES GENERALIZED MASS MATRIX.
C
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/ CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DIMENS/ MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/MASS/    XMASS(50,50)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDO(40)
C
C     ZERO OUT ARRAY.
C
      KDOF=MDOF
      IF(KDOF.LT.6) KDOF=6
      DO 100 INDEX=1,KDOF
      DO 100 INDEX1=1,KDOF
  100 XMASS(INDEX1,INDEX)=0.0
C
C     TRUCK.
C
      DO 200 J=1,MASSES
  200 XMASS(1,1) = XI3(J)    +CGMASS(J)*(XBARCM(1,J)**2+XBARCM(2,J)**2)+
     1                XMASS(1,1)
      XMASS(1,5) = -XIYZ
      DO 300 J=1,MASSES
      XMASS(1,5)=XMASS(1,5)-CGMASS(J)*XBARCM(2,J)*XBARCM(3,J)
      XMASS(2,2)=XMASS(2,2)+CGMASS(J)
      XMASS(4,4)=XMASS(4,4) +XI1(J)  +CGMASS(J)*(XBARCM(2,J)**2+
     1                XBARCM(3,J)**2)
  300 XMASS(5,5)=XMASS(5,5) +XI2(J)  +CGMASS(J)*(XBARCM(1,J)**2+
     1                XBARCM(3,J)**2)
      XMASS(3,3)=XMASS(2,2)
      XMASS(6,6)=XMASS(2,2)
      IF (MASSES.EQ.1) GO TO 700
C
C     AXLES.
C
      KDOF=6
      DO 400 INDEX=1,KAXLES
      M=INDEX+1
      KDOF=KDOF+1
C
C     PSI.
C
      XMASS(1,KDOF) = XI3(M)
      XMASS(KDOF,KDOF) = XI3(M)
C
C     Y.
C
      KDOF=KDOF+1
      XMASS(1,KDOF)=CGMASS(M)*XBARCM(1,M)
      XMASS(3,KDOF)=CGMASS(M)
      XMASS(4,KDOF)=-CGMASS(M)*XBARCM(3,M)
  400 XMASS(KDOF,KDOF)=CGMASS(M)
C
C     CHECK FOR SHELTER.
C
      IF (KSHELT.EQ.0) GO TO 500
      M=M+1
C
C     PSI.
C
      KDOF=KDOF+1
      XMASS(1,KDOF) = XI3(M)
      XMASS(KDOF,KDOF) = XI3(M)
C
```

```
C     Y.
C
      KDOF=KDOF+1
      XMASS(1,KDOF)=CGMASS(M)*XBARCM(1,M)
      XMASS(3,KDOF)=CGMASS(M)
      XMASS(4,KDOF)=-CGMASS(M)*XBARCM(3,M)
      XMASS(KDOF,KDOF)=CGMASS(M)
C
C     PHI.
C
      KDOF=KDOF+1
      XMASS(4,KDOF) = XI1(M)
      XMASS(KDOF,KDOF) = XI1(M)
C
C     Z.
C
      KDOF=KDOF+1
      XMASS(4,KDOF)=CGMASS(M)*XBARCM(2,M)
      XMASS(5,KDOF)=-CGMASS(M)*XBARCM(1,M)
      XMASS(5,KDOF)=CGMASS(M)
      XMASS(KDOF,KDOF)=CGMASS(M)
C
C     RACKS.
C
  500 IF (KRACKS.EQ.0) GO TO 700
      KX=2*KRACKS
      DO 600 INDEX=1,KX
      M=M+1
C
C     PSI.
C
      KDOF=KDOF+1
      XMASS(1,KDOF) = XI3(M)
      XMASS(KDOF,KDOF) = XI3(M)
C
C     X.
C
      KDOF=KDOF+1
      XMASS(1,KDOF)=-CGMASS(M)*XBARCM(2,M)
      XMASS(2,KDOF)=CGMASS(M)
      XMASS(5,KDOF)=CGMASS(M)*XBARCM(3,M)
      XMASS(KDOF,KDOF)=CGMASS(M)
C
C     Y.
C
      KDOF=KDOF+1
      XMASS(1,KDOF)=CGMASS(M)*XBARCM(1,M)
      XMASS(3,KDOF)=CGMASS(M)
      XMASS(4,KDOF)=-CGMASS(M)*XBARCM(3,M)
      XMASS(KDOF,KDOF)=CGMASS(M)
C
C     PHI.
C
      KDOF=KDOF+1
      XMASS(4,KDOF) = XI1(M)
      XMASS(KDOF,KDOF) = XI1(M)
C
C     THETA.
C
      KDOF=KDOF+1
      XMASS(5,KDOF) = XI2(M)
      XMASS(KDOF,KDOF)=XI2(M)
C
C     Z.
C
      KDOF=KDOF+1
      XMASS(5,KDOF)=-CGMASS(M)*XBARCM(1,M)
      XMASS(4,KDOF)=CGMASS(M)*XBARCM(2,M)
      XMASS(5,KDOF)=CGMASS(M)
  600 XMASS(KDOF,KDOF)=CGMASS(M)
C
C     SYMMETRY.
C
```

```
 700  M=MDOF-1
      DO 800 I=1,M
      J1=I+1
      DO 800 J=J1,MDOF
  800 XMASS(J,I)=XMASS(I,J)
C
      RETURN
      END
```

MASSM

```
      SUBROUTINE MATXIN (A, NZ, MAXZ, INDEX)
C     THIS SUBROUTINE WILL INVERT ANY MATRIX (MAXIMUM ORDER OF  50) BY A
C       MODIFIED GAUSS-ELIMINATION METHOD.
C          A  = THE INPUT MATRIX AS WELL AS THE OUTPUT MATRIX.
C         NZ  = THE ORDER OF MATRIX A.
C       MAXZ  = THE MAXIMUM ORDER DIMENSIONED IN THE CALLING PROGRAM.
C     INDEX   = 1 IF THE INVERSE IS FOUND.
C             = 2 IF THE INPUT MATRIX IS SINGULAR.
C             = 3 IF MACHINE ERROR OCCURRED. IF PROGRAMMER WISHES TO LOOP
C                 BACK FOR ANOTHER TRY, BE SURE TO RESET THE INPUT MATRIX.
C     MATRIX A IS DUMMY DIMENSIONED. THIS SUBROUTINE REFERS TO IT AS A
C       SINGLE DIMENSIONED VARIABLE BY FINDING THE PROPER SUBSCRIPT.
C
      COMMON/MATDIM/ KOL(50),ROW(50)
      DIMENSION A(1)
C
      N=NZ
      MAX=MAXZ
      KOL(1)=1
      DO 10 I=2,N
      KOL(I)=KOL(I-1)+1
   10 CONTINUE
      DO 120 K=1,N
      L=N-K+1
      M=KOL(1)
      J=1
      IF(N-K) 190,60,20
   20 AMPY= ABS(A(1))
      DO 40 I=2,L
      IF (AMPY- ABS(A(I))) 30,40,40
   30 J=I
      AMPY= ABS(A(I))
      M=KOL(I)
   40 CONTINUE
      IF(KOL(1)-M) 50,60, 50
   50 KOL(J)=KOL(1)
      KOL(1)=M
   60 IF(A(J)) 70,200,70
   70 AMPY=A(J)
      DO 80 I=2,N
      IS=(I-1)*MAX+J
      ROW(I-1)=A(IS)/AMPY
      IC=(I-2)*MAX
      IS=IC+J
      IT=IC+1
      A(IS)=A(IT)
   80 CONTINUE
      ROW(N)=1.0/AMPY
      IC=(N-1)*MAX
      IS=IC+J
      IT=IC+1
      A(IS)=A(IT)
      DO 100 I=2,N
      AMPY=A(I)
      DO 90 J=2,N
      IS=(J-2)*MAX+I-1
      IT=(J-1)*MAX+I
      A(IS)=A(IT)-AMPY*ROW(J-1)
   90 CONTINUE
      IS=(N-1)*MAX+I-1
      A(IS)= -AMPY*ROW(N)
  100 CONTINUE
      DO 110 J=1,N
      KOL(J)=KOL(J+1)
      IS=(J-1)*MAX+N
      A(IS)=ROW(J)
  110 CONTINUE
      KOL(N)=M
  120 CONTINUE
      DO 170 K=1,N
      IF(KOL(K)-K) 190,170,130
  130 DO 160 I=K,N
      IF(KOL(I)-K) 190,140,150
```

```
140 DO 150 J=1,N
    IS=(I-1)*MAX+J
    IT=(K-1)*MAX+J
    ROW(1)=A(IS)
    A(IS)=A(IT)
    A(IT)=ROW(1)
150 CONTINUE
    M=KOL(K)
    KOL(K)=KOL(I)
    KOL(I)=M
    GO TO 170
160 CONTINUE
    INDEX=3
    GO TO 180
170 CONTINUE
    INDEX=1
180 RETURN
190 INDEX=3
    GO TO 180
200 INDEX=2
    GO TO 180
    END
```

MATXIN

```
      SUBROUTINE MAXI
C
C     RECORD MAXIMUM AND MINIMUM VALUES.
C
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/ CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1    DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2    XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DELTAS/ DISP(100)
      COMMON/MAXMIN/ AMAX(50),AMIN(50),ANGLE(19),CIP(19,6),
     1    CIQ(19,6),CPSIQ(19,6),CPSO(19,6),CRKM(19,5),DMAX(50),
     2    DMIN(50),IFL(19,6),JMAX,KMAX,SMAX(100),SMIN(100),
     3    TAMAX(50),TAMIN(50),TDMAX(50),TDMIN(50),TSMAX(100),
     4    TSMIN(100),TVMAX(50),TVMIN(50),VMAX(50),VMIN(50),
     5    YIELD(6)
      COMMON/MOVING/ XOSAVE,XOBAR,XSAVE,XTB(5),YOSAVE,YOBAR,YSAVE,
     1    ZOSAVE,ZOBAR,ZSAVE,ZTB(5)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IDD(40)
C
      DO 100 INDEX=1,MDOF
      IF (GENDIS(INDEX).LE.DMAX(INDEX)) GO TO 10
      DMAX (INDEX)=GENDIS(INDEX)
      TDMAX(INDEX)=TIME
   10 IF (GENDIS(INDEX).GE.DMIN(INDEX)) GO TO 20
      DMIN (INDEX)=GENDIS(INDEX)
      TDMIN(INDEX)=TIME
   20 IF (GENVEL(INDEX).LE.VMAX(INDEX)) GO TO 30
      VMAX (INDEX)=GENVEL(INDEX)
      TVMAX(INDEX)=TIME
   30 IF (GENVEL(INDEX).GE.VMIN(INDEX)) GO TO 40
      VMIN (INDEX)=GENVEL(INDEX)
      TVMIN(INDEX)=TIME
   40 IF (GENACC(INDEX).LE.AMAX(INDEX)) GO TO 50
      AMAX (INDEX)=GENACC(INDEX)
      TAMAX(INDEX)=TIME
   50 IF (GENACC(INDEX).GE.AMIN(INDEX)) GO TO 100
      AMIN (INDEX)=GENACC(INDEX)
      TAMIN(INDEX)=TIME
  100 CONTINUE
C     OVERTURNING DATA.
C
      IF (TIME .GT.0.0) GO TO 1050
      D = 1000.
      DVEL = 0.0
      DDMIN = 1000.
      TMIN = 0.0
      DCG = 0.0
 1050 IF (DVEL.GT.0.0) GO TO 1500
      DP=D
      IF (MASSES.GT.1) GO TO 1070
      DD1 = XTB(1) - XOBAR
      DD2 = XOBAR - XTB(2)
      D = AMIN1(DD1,DD2)
      GO TO 1150
 1070 D = 1000.
      DO 1100 I=1,4
      DX=XTB(I+1)-XTB(I)
      DZ=ZTB(I+1)-ZTB(I)
      DD=(DZ*(XTB(I+1)-XOBAR)-DX*(ZTB(I+1)-ZOBAR))
     1    /SQRT(DX**2+DZ**2)
      IF (DD.LT.D) D=DD
 1100 CONTINUE
 1150 DCG = D
      IF (D.GE.DDMIN) GO TO 1200
      DDMIN = D
      TMIN = TIME
      IF (TIME.EQ.0.) DCG0 = D
 1200 IF (D.LE.0.0) DVEL = AMAX1((DP-D)/DELTIM,0.001)
```

```
 1500 IF (MASSES.EQ.1) RETURN
C
      DO 220 INDEX=1,MSPRNG
      IF (DISP(INDEX).LE.SMAX(INDEX)) GO TO 210
      SMAX (INDEX)=DISP(INDEX)
      TSMAX(INDEX)=TIME
  210 IF (DISP(INDEX).GE.SMIN(INDEX)) GO TO 220
      SMIN (INDEX)=DISP(INDEX)
      TSMIN(INDEX)=TIME
  220 CONTINUE
C
      RETURN
      END
```

MAXI

```
      SUBROUTINE MOTION
C
C     SUBROUTINE MOTION IS EXECUTIVE RESPONSE PROGRAM.
C
      COMMON/ADAM/    KSTART,NEQ,NIT,QR(100),QRD(6,100),QRP(100),TIO12,
     1    TIO2,TIO24,TIO72
      COMMON/BLAST/   A,AD,AD1,AD2,ALT,A0,COSA,DELTA,KBLAST,PIMP,PR,
     1    PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/CG/      XCG,YCG,ZCG
      COMMON/CHECK/   JDEBUG
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1    DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2    XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DELTAS/  DISP(100)
      COMMON/DIMENS/  MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/EULC/    PDSAV,PSAV,RDSAV,RSAV,YDSAV,YSAV
      COMMON/INTFLG/  IFIRST
      COMMON/LOAD/    AL(6),A6(6),CK(6),CP(6),DDCODE(5),
     1    EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2    NPS(6,4),PI,PSINF(6),PS00(6),S(16,6,4),SIG(5),
     3    SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,6,4),
     4    SJ1(16,6,4),SJ2(16,6,4),SJ3(16,6,4),SJ4(16,6,4),
     5    TD(16,6,4),WG(6),XBOX(16,6,4),YBOX(16,6,4),
     5    ZBOX(16,6,4), TDLAST
      COMMON/MASS/    XMASS(50,50)
      COMMON/MOVING/  XDSAVE,XOBAR,XSAVE,XTB(5),YDSAVE,YOBAR,YSAVE,
     1    ZDSAVE,ZOBAR,ZSAVE,ZTB(5)
      COMMON/MAXMIN/  AMAX(50),AMIN(50),ANGLE(19),CIP(19,6),
     1    CIQ(19,6),CPSIQ(19,6),CPS0(19,6),CRKM(19,6),DMAX(50),
     2    DMIN(50),IFL(19,6),JMAX,KMAX,SMAX(100),SMIN(100),
     3    TAMAX(50),TAMIN(50),TDMAX(50),TDMIN(50),TSMAX(100),
     4    TSMIN(100),TVMAX(50),TVMIN(50),VMAX(50),VMIN(50),
     5    YIELD(6)
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IDD(40)
C
      DO 90  INDEX=1,MDOF
      DMAX(INDEX)=-1000.
      VMAX(INDEX)=-10.0E10
      AMAX(INDEX)=-10.0E10
      DMIN(INDEX)=1000.0
      VMIN(INDEX)=10.0E10
   90 AMIN(INDEX)=10.0E10
      DO 91  INDEX=1,MSPRNG
      SMAX(INDEX)=-10.0E10
   91 SMIN(INDEX)=10.0E10
C
C
C     SET UP ORIGIN OF MOVING COORDINATE SYSTEM.
C
      XDSAVE=0.0
      YDSAVE=0.0
      ZDSAVE=0.0
      RSAV=ROLL
      PSAV=PITCH
      YSAV=YAW
      RDSAV=0.0
      PDSAV=0.0
      YDSAV=0.0
      IFIRST=0
C
C     INITIALIZE INTEGRATION PACKAGE.
C
      DELTIM = DELTX1
      DPRT = -0.5*DELTIM - DELTX
      CALL ADAMI(DELTIM)
C
C     SET TRIM ACCELERATIONS TO ZERO.
```

```
C
      DO 100 INDEX=1,MDOF
      GENVEL(INDEX)=0.0
  100 GENACC(INDEX)=0.0
      IF (KTIRE.GT.1) GO TO 140
      KTIRE = 2
C
C     SET UP MASS MATRIX.
C
      CALL MASSM
      IF (JDEBUG.EQ.0) GO TO 120
      WRITE (6,10)
      DO 110 INDEX=1,MDOF
  110 WRITE (6,7) INDEX,(XMASS(IO,INDEX),IO=1,MDOF)
C
C     INVERT MASS MATRIX.
C
  120 KDUM=0
      CALL MATXIN (XMASS,MDOF,MAXDOF,KDUM)
      IF (KDUM.EQ.1) GO TO 140
      IF(KDUM.EQ.2) WRITE(6,8)
      IF(KDUM.EQ.3) WRITE(6,9)
      KERR = 2
      RETURN
C
C     HERE WE GO...
C
  140 WRITE (6,5)
  180 NOUT = -1
      IF (NPRINT.EQ.0.OR.TIME.LT.DPRT) GO TO 200
      DPRT = DPRT + DPRT1
      NOUT = 0
  200 CONTINUE
      CALL ADAM5(GENDIS,GENVEL,GENACC,TIME,DELTIM)
      IF (NIT.EQ.3) GO TO 260
      IF (KSTART.EQ.6.AND.IFIRST.EQ.1) GO TO 260
      GO TO 280
  260 CONTINUE
      CALL EULER(DELTIM)
      CALL BETAIJ
      CALL ORIGIN(DELTIM)
  280 CONTINUE
      IF (MDOF.LE.6) GO TO 285
      IF (NIT.EQ.1) CALL BOTTOM
  285 CONTINUE
C
C     CALC. NEW FORCES,INCLUDING DAMPING.
C
      CALL EXTRNL
      IF (KSTART.EQ.6) IFIRST = 1
C
C     MULTIPLY MASS-INVERSE INTO R.H.S.
C
      CALL MULT (XMASS,FORCE,GENACC,MAXDOF,MAXDOF,MDOF,MDOF,1)
C
C     IF NIT:2, STILL IN START UP PHASE.
C
      IF (NIT.GT.2) GO TO 200
C
      CALL MAXI
C
C     OUTPUT DATA.
C
      IF(NOUT.NE.0) GO TO 400
      WRITE (6,1)  TIME,DCG
      WRITE (6,2)  (GENDIS(IO),IO=1,MDOF)
      WRITE (6,3)  (GENVEL(IO),IO=1,MDOF)
      WRITE (6,4)  (GENACC(IO),IO=1,MDOF)
      WRITE (6,11) ROLL,PITCH,YAW,XOBAR,YOBAR,ZOBAR
      IF (JDEBUG.EQ.2) WRITE(6,6) (DISP(IO),IO=1,MSPRNG)
C     CHECK FOR PLOT DATA TAPE.
      IF(JGRAPH.GT.0) WRITE (8) TIME,DCG,(GENDIS(IO),IO=1,MDOF),
     1                (GENVEL(IO),IO=1,MDOF),(GENACC(IO),IO=1,MDOF),ROLL,
                                                                  MOTION
```

```
     2                     PITCH,YAW,XOBAR,YOBAR,ZOBAR,
     3                     (DISF(IO),IO=1,MSPRNG)
C
  400 IF (DVEL.GT.0.0.AND.KDAM.GT.0) GO TO 500
      IF (TIME.GE.ENDTIM) GO TO 500
      IF (NIT.GT.1) GO TO 180
      IF (TIME.LT.ENDTX.OR.TIME.GT.ENDTX + DELTX1) GO TO 180
      DELTIM = DELTX
      NIT = 2
      KSTART = 2
      IFIRST = 0
      TIO2 = DELTIM/2.
      TIO12 = DELTIM/12.
      TIO24 = DELTIM/24.
      TIO72 = DELTIM/720.
      GO TO 180
C
  500 WRITE (6,19)
      RETURN
C
C     FORMAT STATEMENTS.
C
    1 FORMAT (7HOTIME =E14.4,10X,14HCG DISTANCE = E15.6)
    2 FORMAT (15HODISPLACEMENT =,/,(1X,10E12.4))
    3 FORMAT (11HOVELOCITY =,/,(1X,10E12.4))
    4 FORMAT (15HOACCELERATION =,/,(1X,10E12.4))
    5 FORMAT (33HOBEGIN TIME HISTORY CALCULATIONS.,/)
    6 FORMAT (21HOSPRING DEFLECTIONS =,/,(1X,10E12.4))
    7 FORMAT (6H COL.  ,I3,(10E12.4))
    8 FORMAT (10(2H *),30HMASS MATRIX SINGULAR. (MOTION))
    9 FORMAT (10(2H ?),43HNUMBER OF DEGREES OF FREEDOM LT 1. (MOTION))
   10 FORMAT (1H0,5(2H *),22H MASS MATRIX BY COLUMN,/)
   11 FORMAT (7HOROLL =,E13.5,9H, PITCH =,E13.5,7H, YAW =,E13.5,
    1   8H,    X = E12.5,6H, Y = E12.5,6H, Z = E12.5)
   17 FORMAT (1H0,10(2H* ),17HNORMAL END OF JOB,10(2H *))
   18 FORMAT (48HOVEHICLE OVERTURNED, RESPONSE TERMINATED AT T = E15.6)
C
   19 FORMAT (24HOTIME HISTORY COMPLETED.)
      END
                                                                MOTION
```

```
      SUBROUTINE MULT (A,B,R,N,M,NN,MM,LL)
C
C        GENERAL MATRIX MULTIPLICATION.
C
C     PARAMETERS -
C     A = NAME OF FIRST INPUT MATRIX.
C     B = NAME OF SECOND INPUT MATRIX.
C     R = NAME OF OUTPUT MATRIX.
C     N = NUMBER OF ROWS IN A.
C     M = NUMBER OF COLUMNS IN A AND ROWS IN B.
C     NN = LENGTH OF COLUMN IN A.
C     MM = LENGTH OF ROW IN A.
C     LL = LENGTH OF ROW IN B.
C
C     REMARKS -
C
C     1. ALL MATRICES MUST BE STORED AS GENERAL MATRICES.
C     2. MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A.
C     3. MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B.
C     4. NUMBER OF COLUMNS OF MATRIX A MUXT BE EQUAL TO THE NUMBER
C        OF ROWS OF MATRIX B.
C     5. NN.LE.N, MM.LE.M.
C
C     *****     *****
C
      DIMENSION A(1),B(1),R(1)
C
      IK=-M
      DO 20 K = 1,LL
      IK=IK+M
      IRO = N*(K-1)
       DO 20 J=1,NN
       IR = IRO + J
       JI=J-N
       IB = IK
       RIR = 0.
        DO 10 I=1,MM
         JI=JI+N
        IB = IB + 1
   10   RIR = RIR + A(JI)*B(IB)
   20 R(IR) = RIR
C
      RETURN
      END
```

MULT

```
      SUBROUTINE ORIGIN(DELTAT)
C
C     DETERMINE POSITION VECTOR OF THE ORIGIN IN FIXED AXIS SYSTEM.
C
      COMMON/CALC/    BETA (3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50), GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/MOVING/ XDSAVE,XOBAR,XSAVE,XTB(5),YDSAVE,YOBAR,YSAVE,
     1    ZDSAVE,ZOBAR,ZSAVE,ZTB(5)
C
C
      XOBARD=BETA(1,1)*GENVEL(2)+BETA(1,2)*GENVEL(3)+BETA(1,3)*GENVEL(6)
      YOBARD=BETA(2,1)*GENVEL(2)+BETA(2,2)*GENVEL(3)+BETA(2,3)*GENVEL(6)
      ZOBARD=BETA(3,1)*GENVEL(2)+BETA(3,2)*GENVEL(3)+BETA(3,3)*GENVEL(6)
C
C     TRAPEZOIDAL INTEGRATION.
C
      XOBAR    =XSAVE+0.5*(XDSAVE+XOBARD)*DELTAT
      YOBAR    =YSAVE+0.5*(YDSAVE+YOBARD)*DELTAT
      ZOBAR    =ZSAVE+0.5*(ZDSAVE+ZOBARD)*DELTAT
C
C     SAVE VALUES FOR NEXT STEP
C
      XSAVE=XOBAR
      YSAVE=YOBAR
      ZSAVE=ZOBAR
      XDSAVE=XOBARD
      YDSAVE=YOBARD
      ZDSAVE=ZOBARD
C
      RETURN
      END
```

ORIGIN

```
      SUBROUTINE PITER (C,R,NTRIAL,KOK)
C
C     C - CRITERIA (C .LT. 1)
C         NEGATIVE VALUE INDICATES UNSTABLE RESPONSE.
C         CRITICAL VALUE OF C = 0.
C     R - PRESSURE.
C     KOK  - 0, ANOTHER TRIAL IS REQUIRED
C            1, ITERATION COMPLETE
C            2, ERROR.
C
      DIMENSION R(5),C(5)
      DATA EX1/.25/, EX2/2.0/, FMIN/0.5/, FMAX/2.0/
      DATA D1/0.5/, D2/0.2/, TOL1/0.2/, TOL2/0.03/
C
      RT2(A1,A2,B1,B2) = (A1*B2-A2*B1)/(B2-B1)
C
      IF (C(1).GT.1.0) GO TO 3300
      IF (C(1).LT.0.0) C(1) = -1.0
      IF (NTRIAL - 2) 100,600,1200
C
C     ONE TRIAL COMPLETE.
C
  100 DO 200 I=3,5
      R(I) = 0.0
  200 C(I) = 0.0
      R(5) = 1.E11
      R(2) = R(1)
      C(2) = C(1)
      F = FMIN
  300 IF (C(2).LE.0.0) GO TO 500
      F1 = (1.0/(1.0-C(2)**EX2))**EX1
      F = AMIN1(FMAX,F1)
  500 R(3) = F*R(2)
      R(1) = R(3)
      IF (ABS(C(2)).LT.TOL1) GO TO 3200
      GO TO 3100
C
C     TWO TRIALS COMPLETED.
C
  600 C(3) = C(1)
      R(3) = R(1)
      IF (C(3).LE.0.0) GO TO 700
      IF (C(2) .LE.0.0) GO TO 800
C     BOTH POINTS GOOD.
      IF ((R(2) - R(3))*(C(2) - C(3)).LT.0.0) GO TO 650
      F  = (1.0/(1.0-C(3)**EX2))**EX1
      F = AMIN1(FMAX,F)
      R(4) = F*R(3)
      R(1) = R(4)
      GO TO 3100
  650 R(4) = RT2(R(2),R(3),C(2)**2,C(3)**2)
      R(4) = AMIN1(R(4),FMAX*AMAX1(R(2),R(3)))
      R(1) = R(4)
      IF (ABS(C(3))        .LT.TOL1) GO TO 3200
      IF (ABS(R(4) - R(3))/R(4).LE.TOL2) GO TO 3200
      GO TO 3100
  700 IF (C(2).LE.0.0) GO TO 1100
C     MOST RECENT POINT BAD.
      R(4) = R(2) - D1*(R(2) - R(3))
      R(1) = R(4)
      IF (ABS(R(4) - R(2))/R(4).LT.TOL2) GO TO 3200
      GO TO 3100
C     MOST RECENT GOOD, OTHER BAD.
  800 R(4) = R(3) - D1*(R(3) - R(2))
      R(1) = R(4)
      IF (ABS(C(3))        .LT.TOL1) GO TO 3200
      IF (ABS(R(4) - R(3))/R(4).LT.TOL2) GO TO 3200
      GO TO 3100
C     BOTH POINTS BAD.
 1100 R(4) = FMIN*R(3)
      R(1) = R(4)
      GO TO 3100
C
```

```
C      THREE OR MORE POINTS COMPLETED.
C
 1200 C(4) = C(1)
      R(4) = R(1)
      IF (C(4).GT.0.0) GO TO 1400
C     CURRENT POINT BAD.
      IF (C(3).GT.0.0) GO TO 1350
      IF (C(2).GT.0.0) GO TO 1250
C     ALL POINTS BAD, KEEP 3 AND 4.
      C(2) = C(3)
      R(2) = R(3)
      C(3) = C(4)
      R(3) = R(4)
      R(4) = FMIN*R(4)
      R(1) = R(4)
      GO TO 3100
C     POINTS 3 AND 4 BAD, KEEP 2 AND 4.
 1250 C(3) = C(4)
      R(3) = R(4)
      GO TO 700
C     KEEP 3 AND 4.
 1350 C(2) = C(3)
      R(2) = R(3)
      C(3) = C(4)
      R(3) = R(4)
      GO TO 700
C     CURRENT POINT GOOD.
 1400 IF (C(3).GT.0.0) GO TO 1900
      IF (C(2).GT.0.0) GO TO 1500
C     BOTH PREVIOUS POINTS BAD, KEEP POINTS 3 AND 4.
 1450 C(2) = C(3)
      R(2) = R(3)
      C(3) = C(4)
      R(3) = R(4)
      GO TO 800
C     POINTS 2 AND 4 ARE GOOD, 3 NO GOOD.
 1500 IF ((R(2) - R(4))*(C(2) - C(4)).GT.0.0) GO TO 1450
C     KEEP 2 AND 4.
      R3 = R(3) + D2*(R(4) - R(3))
      R(5) = R(3)
 1800 R(3) = R(4)
      C(3) = C(4)
      R(4) = RT2 (R(2),R(3),C(2)**2,C(3)**2)
      R(4) = AMIN1(R(4),R3)
      R(1) = R(4)
      IF (ABS(C(3))        .LT.TOL1) GO TO 3200
      IF (ABS(R(4) - R(3))/R(4).LT.TOL2) GO TO 3200
      GO TO 3100
C
 1900 IF (C(2).GE.0.0) GO TO 2100
      IF ((R(3)-R(4))*(C(3)-C(4)).LT.0.0) GO TO 2000
      R(3) = R(4)
      C(3) = C(4)
      GO TO 800
C     KEEP POINTS 3 AND 4.
 2000 R3 = R(2) + D2*(R(4) - R(2))
      R(5) = R(2)
      C(2) = C(3)
      R(2) = R(3)
      GO TO 1800
C
C     ALL THREE POINTS ARE GOOD.
C
C     ALL 3 POINTS ARE GOOD.
 2100 IF ((R(3)-R(4))*(C(3)-C(4)).LT.0.0) GO TO 2200
      F   = (1.0/(1.0-C(4)**EX2))**EX1
      F = AMIN1(FMAX,F)
      R1 = F*R(4)
      GO TO 2300
 2200 R1 = RT2 (R(3),R(4),C(3)**2,C(4)**2)
      R1 = AMIN1(R1,FMAX*R(4))
 2300 R(2) = R(3)
      C(2) = C(3)
```

```
      R(3) = R(4)
      C(3) = C(4)
C     CHECK EARLIER BAD POINT.
      IF (R1.GT.R(5)) R1 = R(5) + D2*(R(4) - R(5))
      R(4) = R1
      R(1) = R(4)
      IF (ABS(C(4))          .LT.TOL1) GO TO 3200
      IF (ABS(R(4)-R(3))/R(4).LE.TOL2) GO TO 3200
C
C     NEW TRIAL REQUIRED.
C
 3100 KOK = 0
      RETURN
C
C     SOLUTION FOUND.
C
 3200 KOK = 1
      RETURN
C
C     ERROR.
C
 3300 KOK = 2
      WRITE (6,5000) C(1)
 5000 FORMAT (44H0ERROR IN S/R RITER - CRIT GREATER THAN 1.0, E15.6)
      RETURN
C
      END
                                                                PITER
```

```
      SUBROUTINE PRETRM
C
C     CALLED ONLY IF GUY WIRES ARE USED.
C
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/CHECK/  JDEBUG
      COMMON XLAMDA(100,44)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IOD(40)
      COMMON/PRELOD/ PRELOD(4)
      COMMON/SPRING/ FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1   X(600)
      COMMON/WEIGHT/ WEIGHT,WGTS(13)
      DIMENSION RES( 3),ER( 3),GENSH(3)
C
      WRITE (6,1)
C     FIND GENERALIZED FORCE FOR ONE-HALF SHELTER
C          HEAVE MODE AND PITCH MODE
      IS = 1 +KAXLES +1
      GFGH = -0.5*WGTS(IS)
      GFGH1 = 0.0
      IRL = IS +1
      IRU = IS +KRACKS
      DO 10 IM=IRL,IRU
          GFGH = GFGH -WGTS(IM)
   10     GFGH1 = GFGH1 +WGTS(IM)*(XBARCM(3,IM) -XBARCM(3,IS))
C     FIND GENERALIZED FORCE DUE TO GUY WIRES
      IH=8+2*KAXLES
      IHP1 = IH +1
      IHP2 = IH + 2
      IHM4 = IH - 4
      IHM5 = IH -5
      IHM6=IH-6
      GFCH = 0.0
      GFCH1 = 0.0
      GFCH2 = 0.
      KC1 = 2*KGWIR1 -1
      KC2 = 2*KGWIR2 -1
      I =0
      DO 20 J=KC1,KC2,2
          I = I +1
          GFCH=GFCH-PRELOD(I)*XLAMDA(J,IHM6)
          GFCH1=GFCH1-PRELOD(I)*XLAMDA(J,IHM5)
   20     GFCH2 = GFCH2 - PRELOD(I)*XLAMDA(J,IHM4)
      GFGCH = GFGH +GFCH
      GFGCH1 = GFGH1 +GFCH1
      GFGCH2 = GFCH2
C     START RELAXATION PROCESS
      NCOUNT =0
      NPR = 0
      IF (JDEBUG.EQ.2) NPR = 1
      NTRY =0
      DO 25 I=1,MDOF
   25 GENDIS(I)=0.0
      DO 30 I=1,3
          ER(I)=0.0001
   30     RES(I) = 0.0
C
C     SET STARTING POSITION FOR TRIM.
C
      AVSLO=0.0
      ISK=KBOG+1
      DO 301 INDEX=1,KAXLES
  301 ISK=ISK+NASPR(INDEX)
      ISKK=ISK
      I = 2*(ISK + KBOG) - 1
      IK = I
      DO 33 J=1,KSHELT
      IS=JCURVE(ISK)
```

```
      NSTOP=NPPC(ISK)+IS-1
   31 IS=IS+1
      IF (IS.GT.NSTOP) GO TO 2000
      IF (K(IS).LT.0.0) GO TO 31
      AVSLO = AVSLO - ((FOFX(IS-1) - FOFX(IS))/
     1  (X(IS-1) - X(IS)))*XLAMDA(I,IHM6)**2
      I = I + 2
      ISK=ISK+1
   33 CONTINUE
      GENDIS(IH)=GFGCH/AVSLO
      GENSH(1)=GENDIS(IH)
      GENSH(2)=0.0
      GENSH(3)=0.
   35 CONTINUE
      GFSH1 = 0.0
      GFSH = 0.0
      GFSH2 = 0.
      ISK=ISKK
      I = IK
      DO 40 J=1,KSHELT
           DISPJ=XLAMDA(I,IHM6)*GENDIS(IH)+XLAMDA(I,IHM5)*
     1                GENDIS(IHP1) + XLAMDA(I,IHM4)*GENDIS(IHP2)
           ISTART = JCURVE(ISK)
           NP = NPPC(ISK)
           CALL INT1(DISPJ,NP,X(ISTART),FOFX(ISTART),SF)
           GFSH=GFSH+SF*XLAMDA(I,IHM6)
           GFSH1=GFSH1+SF*XLAMDA(I,IHM5)
           GFSH2 = GFSH2 + SF*XLAMDA(I,IHM4)
           I = I +2
           ISK = ISK +1
   40 CONTINUE
      RES( 1) = GFSH +GFGCH
      RES(   2) = GFSH1 +GFGCH1
      RES(3) = GFSH2 + GFGCH2
      CALL RELAX (3,RES,GENSH,ER,NOK,NPR,NCOUNT)
      GENDIS(IH) = GENSH(1)
      GENDIS(IHP1) = GENSH(2)
      GENDIS(IHP2) = GENSH(3)
      IF(NCOUNT.EQ.1) NTRY = NTRY +1
      IF(NTRY.GT.20) GO TO 1000
      IF(NOK.EQ.0) GO TO 35
      IF(NOK.EQ.2) GO TO 1000
C     PRE TRIM DONE
      IF (KRACKS.EQ.0) GO TO 60
C     DISPLACE RACKS.
      IH = IH + 5
      DO 50 I=1,KRACKS
      GENDIS(IH) = GENSH(1)
      GENDIS(IH+6*KRACKS) = GENSH(1)
      GENDIS(IH+3) = GENSH(3)
      GENDIS(IH+3+6*KRACKS) = GENSH(3)
   50 IH = IH + 6
   60 WRITE (6,2)
      IF (JDEBUG.GT.0) WRITE (6,4) GENDIS
      CALL BASE
      RETURN
 1000 WRITE(6,9500)
      KERR = 2
      RETURN
C
C     ERROR IN DATA ?
C
 2000 WRITE (6,3)
      KERR  = 3
      RETURN
C
C     FORMAT STATEMENTS.
C
    1 FORMAT (28H BEGIN PRE-TRIM ON GUY WIRES)
    2 FORMAT (18H PRETRIM COMPLETED)
    3 FORMAT (38H ERROR DETECTED BY CABLE TRIM ROUTINE.,/,
     1        40H CHECK SHELTER X VS. F(X) DATA. (PRETRM))
    4 FORMAT (10H0GENDIS = /(1X,10E13.4))
```

PRETR

```
9500 FORMAT (/,31H GUY WIRE TRIM ABORTED (PRETRM))
     END
```

PRETR

```
      SUBROUTINE REBOUN (NSP)
C
C     SUBROUTINE REBOUN (REBOUND) CALCULATES NEW STARTING
C     CONDITIONS WHEN A SPRING BOTTOMS OUT (REBOUNDS).
C              TOTAL CONSERVATION OF ENERGY IS ASSUMED
C
C     COMMON
C
      COMMON/ADAM/    KSTART,NEQ,NIT,QR(100),QRD(6,100),QRP(100),TIO12,
     1   TIO2,TIO24,TIO72
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCGO,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/DIMENS/  MAXDOF,MAXMAS,MAXSP,MRDIM
      COMMON/EULC/    PDSAV,PSAV,RDSAV,RSAV,YDSAV,YSAV
      COMMON/INTFLG/  IFIRST
      COMMON XLAMDA(100,44)
      COMMON/MASS/    XMASS(50,50)
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDC(40)
C
C        CALCULATE GENERALIZED IMPULSE DUE TO UNIT SPRING FORCE
      DO 110 M=7,MDOF
  110 FORCE(M)=XLAMDA(NSP,M-5)
      IF (KBOG.EQ.0) GO TO 150
      NA=0
      DO 120 I=1,KAXLES
  120 NA=NA+NASPR(I)
      NS=2*NA+1
      IF (NSP.LT.NS) GO TO 150
      IF (NSP.GE.NS+4*KBOG) GO TO 150
      DO 130 M=7,MDOF
  130 FORCE(M)=FORCE(M)+XLAMDA(NSP+2,M-6)
  150 FORCE(1)=0.0
      FORCE(2)=0.0
      FORCE(3)=0.0
      FORCE(4)=0.0
      FORCE(5)=0.0
      FORCE(6)=0.0
C     CALCULATE VELOCITY CHANGE DUE TO UNIT IMPULSE
      CALL MULT (XMASS,FORCE,GENACC,MAXDOF,MAXDOF,MDOF,MDOF,1)
C     CALCULATE VALUE OF IMPULSE NECESSARY TO KEEP KE CONSTANT
      VNUM = 0.
      VDOM = 0.
      DO 170 M=1,MDOF
      VNUM = GENVEL(M)*FORCE(M) + VNUM
  170 VDOM = GENACC(M)*FORCE(M) + VDOM
      VIMP = -2.0*VNUM/VDOM
C     CALCULATE NEW VELOCITIES
      DO 200 M=1,MDOF
         GENVEL(M) = GENVEL(M) +VIMP*GENACC(M)
         GENACC(M) = 0.0
  200 CONTINUE
      RETURN
      END
```

REBOUN

```
      SUBROUTINE RELAX (NEQ,RES,X,ER,NOK,NPRINT,NCOUNT)
C     NCOUNT MUST BE SET  = 0 BEFORE VERY FIRST CALL.
C
C     VERSION 2.0, DECEMBER 1975.
C
      COMMON/RLP/     DELX(50),IP(50),PRES(50),PX(50),RRES(50),
     1   SIGX(50),XRES(50,50),XX1(50)
C
      DIMENSION RES(NEQ),X(NEQ),ER(NEQ)
C
      NDIM=50
      CON=1000.0
      IF (NPRINT.NE.1) GO TO 40
      WRITE (6,50)  (X(N),RES(N),N=1,NEQ)
   50 FORMAT (1H ,//4X,8H TRIAL X,10X,7HRESIDUE/(4X,E13.6,4X,E13.6))
   40 IF (NCOUNT.EQ.0 ) GO TO 10
      IF (NCOUNT.LE.NEQ)  GO TO 14
      NCOUNT=0
      NQ=0
      NNQ=0
      DO   2   I=1,NEQ
      DX    =X(I)-PX(I)
      IF (ABS(DX).LE.ER(I))  NQ=NQ+1
      IF (ABS(RES(I))-ABS(PRES(I)).GT.0.0)   NNQ=NNQ+1
    2 CONTINUE
      IF (NQ.EQ.NEQ)  GO TO 100
      IF (NNQ.EQ.NEQ) GO TO 101
   10 DO 4   I=1,NEQ
      XX1(I)=X(I)
      RRES(I)=RES(I)
      DELX(I)=ABS(0.0001*X(I))
      IF (DELX(I).LT.ER(I))  DELX(I)=ER(I)
    4 CONTINUE
      GO TO 6
   14 DO 7   MM=1,NEQ
    7 XRES(MM,NCOUNT) = (RES(MM) - RRES(MM)) / DELX(NCOUNT)
      X(NCOUNT)=XX1(NCOUNT)
      IF (NCOUNT.EQ.NEQ)  GO TO 8
    6 NCOUNT=NCOUNT+1
      X(NCOUNT)=X(NCOUNT)+DELX(NCOUNT)
      NOK=0
      RETURN
    8 DO 20   I=1,NEQ
   20 SIGX(I)=-RRES(I)
      CALL  SOLVE(XRES,NEQ,NDIM,0,IP,DET,SIGX)
      IF (NEQ.EQ.0)  GO TO 102
      PROP=1.0
      DO 13 I=1,NEQ
      IF(ABS(SIGX(I)).LT.CON*ER(I))  GO TO 13
      XPROP  =  (CON*ER(I))/ABS(SIGX(I))
      IF  (XPROP.LT.PROP)  PROP=XPROP
 13   CONTINUE
      DO 12   I=1,NEQ
      X(I)=XX1(I)+SIGX(I)*PROP
      PX(I)=XX1(I)
   12 PRES(I)=RRES(I)
      NOK=0
      NCOUNT=NEQ+1
      RETURN
  100 NOK=1
      GO TO 11
  101 WRITE  (6,55)
   55 FORMAT (31H0    SOLUTION DIVERGING IN RELAX)
  102 NOK=2
   11 NCOUNT=0
      RETURN
      END
```

RELAX

```
      SUBROUTINE RFORCE(FI,M,KDOF,INDEXY)
C
C
C     RFORCE CALCULATES NON-ACCELERATION INERTIA TERMS FORM RACK MASSES.
C
      COMMON/CALC/    BETA (3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,K3OS,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDC(40)
      DIMENSION FI(1)
C
      FIXJ(XMJ,XLJ,YLJ,ZLJ,GYDOTJ,GZDOTJ)=XMJ*(-GENVEL(1)*GENVEL(3)-XLJ
     1                                  *(GENVEL(1)**2+GENVEL(5)**2)
     2                                  +GENVEL(4)*(YLJ*GENVEL(5)+ZLJ
     3                                  *GENVEL(1))  -2.0*GENVEL(1)*GYDOTJ)
     4                                  +XMJ*GENVEL(5)*(GENVEL(6)+2.0*
     5                                  GZDOTJ)
     4                                  +XMJ*GENVEL(5)*(GENVEL(6)+2.0*
     5                                  GZDOTJ)
      FIYJ(XMJ,XLJ,YLJ,ZLJ,GXDOTJ,GZDOTJ)=XMJ*(GENVEL(1)*GENVEL(2)
     1                                       -YLJ
     1                                  *(GENVEL(1)**2+GENVEL(4)**2)
     2                                  +GENVEL(5)*(XLJ*GENVEL(4)+ZLJ
     3                                  *GENVEL(1))+2.0*GENVEL(1)*GXDOTJ)
     4                                  -XMJ*GENVEL(4)*(GENVEL(6)+2.0*
     5                                  GZDOTJ)
      FIZJ(XMJ,XLJ,YLJ,ZLJ,GXDOTJ,GYDOTJ)=XMJ*(GENVEL(1)*(XLJ*GENVEL(4)
     1          +YLJ*GENVEL(5))-ZLJ*(GENVEL(4)**2+GENVEL(5)**2)+GENVEL(4)
     2          *(GENVEL(3)+2.0*GYDOTJ)-GENVEL(5)*(GENVEL(2)+2.0*GXDOTJ))
      FIPHIJ(J,GDTHET,GDPSI) =  (XI3(J) -XI2(J))*GENVEL(1)
     1                          *GENVEL(5)-(XI1(J) + XI2(J) -XI3(J))*
     2                          GENVEL(1)*GDTHET +(XI1(J) +XI3(J)-XI2(J))
     3                          *GENVEL(5)*GDPSI
      FITHET(J,GDPHI,GDPSI)  =  (XI1(J) -XI3(J))*GENVEL(1)
     1                          *GENVEL(4)+(XI1(J) +XI2(J) -XI3(J))*
     2                          GENVEL(1)*GDPHI -(XI2(J) +XI3(J) -XI1(J))
     3                          *GENVEL(4)*GDPSI
      FIPSI(J,GDPHI,GDTHET)  =  (XI2(J) -XI1(J))*GENVEL(4)
     1                          *GENVEL(5)-(XI1(J) +XI3(J) -XI2(J))*
     2                          GENVEL(5)*GDPHI +(XI2(J) +XI3(J) -XI1(J))*
     3                          GENVEL(4)*GDTHET
C
C
C     RACKS
C
      KX=2*KRACKS
      DO 300 INDEX=1,KX
      FI(5)  =  (XI1(M) -XI3(M) +CGMASS(M)*(XBARCM(3,M)**2
     1          -XBARCM(1,M)**2))*GENVEL(1)*GENVEL(4)+CGMASS(M)
     2          *XBARCM(2,M)*XBARCM(3,M)*GENVEL(4)*GENVEL(5)+2.0
     3          *GENVEL(5)*CGMASS(M)*XBARCM(1,M)*GENVEL(INDEXY-1)
     4          -2.0*GENVEL(1)*CGMASS(M)*XBARCM(3,M)*GENVEL(INDEXY)
     5          -2.0*GENVEL(4)*CGMASS(M)*XBARCM(1,M)*GENVEL(INDEXY)
     6          +GENVEL(1)*(XI1(M) +XI2(M) -XI3(M))
     7          *GENVEL(INDEXY+1)-GENVEL(4)*(XI2(M) +XI3(M) -XI1(M))*
     8          GENVEL(INDEXY-2)+FI(5)
     9          +2.0*GENVEL(5)*CGMASS(M)*XBARCM(3,M)*GENVEL(INDEXY+2)
      FI(6) =FI(6)+GENVEL(4)*(GENVEL(3)*CGMASS(M)+2.0*CGMASS(M)
     1    *GENVEL(INDEXY))-GENVEL(5)*(GENVEL(2)*CGMASS(M)+2.0
     2          *CGMASS(M)*GENVEL(INDEXY-1))
      FI(1)  =  (XI2(M) -XI1(M) +CGMASS(M)*(XBARCM(1,M)**2
     1          -XBARCM(2,M)**2))*GENVEL(4)*GENVEL(5)-CGMASS(M)
     2          *XBARCM(2,M)*XBARCM(3,M)*GENVEL(1)*GENVEL(4)+2.0
     3          *GENVEL(1)*CGMASS(M)*(XBARCM(2,M)*GENVEL(INDEXY)
     4          +XBARCM(1,M)
     5          *GENVEL(INDEXY-1))-GENVEL(5)*(XI1(M) +XI3(M) -XI2(M))*
     6          GENVEL(INDEXY+1)+GENVEL(4)*(XI2(M)  +XI3(M) -XI1(M)
```

RFORCE

```
     7          )*GENVEL(INDEXY+2)+FI(1)
     8          -2.0*GENVEL(4)*CGMASS(M)*XBARCM(1,M)*GENVEL(INDEXY+2)
     9          -2.0*GENVEL(5)*CGMASS(M)*XBARCM(2,M)*GENVEL(INDEXY+2)
      FI(2)  =  -GENVEL(1)*(GENVEL(3)*CGMASS(M)+2.0*CGMASS(M)
     1          *GENVEL(INDEXY))+FI(2)
     2          +GENVEL(5)*(GENVEL(6)*CGMASS(M)+2.0*CGMASS(M)*
     3          GENVEL(INDEXY+2))
      FI(3)  =  GENVEL(1)*(GENVEL(2)*CGMASS(M)+2.0*CGMASS(M)
     1          *GENVEL(INDEXY-1))+FI(3)
     2          -GENVEL(4)*(GENVEL(6)*CGMASS(M)+2.0*CGMASS(M)*
     3          GENVEL(INDEXY+2))
      FI(4)  =  CGMASS(M)*XBARCM(2,M)*XBARCM(3,M)*(GENVEL(1)**2
     1          -GENVEL(5)**2)+(XI3(M) -XI2(M)+CGMASS(M)
     2          *(XBARCM(2,M)**2-XBARCM(3,M)**2))*GENVEL(1)*GENVEL(5)-2.0
     3          *GENVEL(1)*CGMASS(M)*XBARCM(3,M)*GENVEL(INDEXY-1)
     4          -2.0*GENVEL(5)*CGMASS(M)*XBARCM(2,M)*GENVEL(INDEXY-1)
     5          +2.0*GENVEL(4)*CGMASS(M)*XBARCM(2,M)*GENVEL(INDEXY)
     6          -GENVEL(1)*(XI1(M) +XI2(M) -XI3(M))*
     7          GENVEL(INDEXY+2)+GENVEL(5)*(XI1(M) +XI3(M) -XI2(M))*
     8          GENVEL(INDEXY-2)+FI(4)
     9          +2.0*GENVEL(4)*CGMASS(M)*XBARCM(3,M)*GENVEL(INDEXY+2)
      FI(KDOF) = FIPSI(M,GENVEL(INDEXY+1),GENVEL(INDEXY+2))
      KDOF=KDOF+1
      FI(KDOF) = FIXJ(CGMASS(M),XBARCM(1,M),XBARCM(2,M),
     1          XBARCM(3,M),GENVEL(INDEXY),GENVEL(INDEXY+3))
      KDOF = KDOF+1
      FI(KDOF) = FIYJ(CGMASS(M),XBARCM(1,M),XBARCM(2,M),XBARCM(3,M),
     1          GENVEL(INDEXY-1),GENVEL(INDEXY+3))
      KDOF=KDOF+1
      FI(KDOF) = FIPHIJ(M,GENVEL(INDEXY+2),GENVEL(INDEXY-2))
      KDOF=KDOF+1
      FI(KDOF) = FITHET(M,GENVEL(INDEXY+1),GENVEL(INDEXY-2))
      KDOF=KDOF+1
      FI(KDOF)=FIZJ(CGMASS(M),XBARCM(1,M),XBARCM(2,M),XBARCM(3,M),
     1            GENVEL(INDEXY-1),GENVEL(INDEXY))
      KDOF=KDOF+1
      M=M+1
  300 INDEXY=INDEXY+6
C
      RETURN
      END
```

RFORCE

```
      SUBROUTINE SETA
C     THIS S/R SETS UP THE ALPHAS, K, P(0), P(INF).
C     IT ALSO COMPUTES THE SPEED OF THE SHOCK DIFFRACTING AROUND EACH
C     FACE (A).
      COMMON/BLAST/  A,AD,AD1,AD2,ALT,A0,COSA,DELTA,KBLAST,PIMP,PR,
     1   PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/LOAD/   AL(6),A6(6),CK(6),CP(6),DDCODE(6),
     1   EEF(4,6),IIS,IS,JFIR(16,6,4),KDEBUG,KTS(4),NBOX,
     2   NPS(6,4),PI,PSINF(6),PS00(6),S(16,6,4),SIG(6),
     3   SI1(16,6,4),SI2(16,6,4),SI3(16,6,4),SI4(16,6,4),
     4   SJ1(16,6,4),SJ2(16,6,4),SJ3(16,6,4),SJ4(16,6,4),
     5   TJ(16,6,4),WG(6),XBOX(16,6,4),YBOX(16,6,4),
     6   ZBOX(16,6,4), TDLAST
C
      A = AD*PI/180.
      COSA = COS(A)
      SINA = SIN(A)
      AL(1)  = ABS(A)
      AL(2)  = PI-AL(1)
      AL(3)  = PI/2. - A
      AL(4)  = PI-AL(3)
      AL(5)  = PI/2.
      AL(6)  = AL(5)
      A1  = SQRT((8.*Z+7.)*(2.*Z+7.)/(7.*(7.+6.*Z)))
      A2  = SQRT((Z+1.)*(Z+7.)/(6.*Z+7.))
      WW  = 5.0*Z*A0/SQRT(7.*(7.+6.*Z))
      CKK = .1 + .07/Z
      DO 110 I=1,6
      CALL DRAGL(AL(I),CP(I))
      PRB = 0.
      WF = 0.
      AA = 1.
      SIG(I) = (-1.0)**I
      CK(I)  = CKK*(1. - (AL(I)-PI/2.)*0.2*Z/(PI/2.))
      PSINF(I) = PS0
      DDCODE(I)  = 1
      EE  = 1.0
      IF  (AL(I).GT.PI/2. + 1.E-6) GO TO 105
      PSINF(I)  = PS0 + 2.*CP(I)*Q0
      IF  (CP(I).GE.0.0) PSINF(I) = PS0
      CK(I)  = CKK
      AA = A1
      PRB = PR
      EE  = 0.
      WF  = 2.
      IF  (AL(I).LT.PI/4.) GO TO 105
      AA = A1 + (AL(I) - PI/4.)*(A2-A1)/(PI/4.)
      PRB = PR + (1.-PR)*(AL(I)-PI/4.)/(PI/4.)
      WF  = -(AL(I)-PI/4.)/(PI/4.) + 2.0
  105 A6(I)  = A0*AA
      PS00(I)  = PRB*PS0
      WG(I)  = WF*WW*SINA
      IF  (CP(I).LT.0..AND.AL(I).GT.PI/2.+1.E-6) DDCODE(I) = 2
      IF  (AL(I).LT.PI/6.) EE = 1.0-AL(I)*6./PI
      EEF(1,I)  = 1.0
      EEF(2,I)  = 1.0
      EEF(3,I)  = 1.0
      EEF(4,I)  = 1.0
      GO  TO (107,107,108,109,110,110), I
  107 IF  (A.GT.0.) EEF(1,I) = EE
      IF  (A.LT.0.) EEF(3,I) = EE
      GO  TO 110
  108 IF  (A.GE.0.0) EEF(3,I) = EE
      GO  TO 110
  109 IF  (A.LE.0.0) EEF(1,I) = EE
  110 CONTINUE
C     COMPUTE S-BAR.
      DO  140 N=1,NBOX
      DO  140 I=1,6
      NPSS = NPS(I,N)
      IF  (NPSS.EQ.0) GO TO 140
      AA = A6(I) + WG(I)
      AB = A6(I) - WG(I)
```

```
      DO 135 K=1,NPSS
      IF (I.LT.5) GO TO 112
      JFIR(K,I,N) = 1
      GO TO 135
  112 JFIR(K,I,N) = 0
      CC = 1.
      GO TO (115,115,125,120), I
  115 SJ1(K,I,N) = SI1(K,I,N)*(1. - CC*AA*SINA/VS)/AA
      SJ3(K,I,N) = SI3(K,I,N)*(1.+CC*AB*SINA/VS)/AB
      SJ2(K,I,N) = SI2(K,I,N)/A6(I)
      SJ4(K,I,N) = SI4(K,I,N)/A6(I)
      GO TO 135
  120 CC = -1.
  125 SJ1(K,I,N) = SI1(K,I,N)*(1. + CC*AB*COSA/VS)/AB
      SJ3(K,I,N) = SI3(K,I,N)*(1. - CC*AA*COSA/VS)/AA
      SJ2(K,I,N) = SI2(K,I,N)/A6(I)
      SJ4(K,I,N) = SI4(K,I,N)/A6(I)
  135 CONTINUE
  140 CONTINUE
      IF (KDEBUG.GT.0) WRITE (6,1000) EEF
      IF (KDEBUG.GT.0) WRITE (6,1100) WG,A6,PS00,PSINF,CK
 1000 FORMAT (4H0EEF/(4E15.6))
 1100 FORMAT (20H0WG,A6,PS00,PSINF,CK/(6E15.6))
      RETURN
      END
```

SETA

```
      SUBROUTINE SFORCE(FI,M,KDOF,INDEXY)
C
C     SFORCE CALCULATES NON-ACCLERATION INERTIA FORCES FROM
C       SHELTER MASS.
C
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1   DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2   XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MJOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
      DIMENSION FI(1)
C
      FIYJ(XMJ,XLJ,YLJ,ZLJ,GXDOTJ,GZDOTJ)=XMJ*(GENVEL(1)*GENVEL(2)-YLJ
     1                                *(GENVEL(1)**2+GENVEL(4)**2)
     2                                +GENVEL(5)*(XLJ*GENVEL(4)+ZLJ
     3                                *GENVEL(1))+2.0*GENVEL(1)*GXDOTJ)
     4                                 -XMJ*GENVEL(4)*(GENVEL(6)+2.0
     5                                *GZDOTJ)
      FIPHIJ(J,GDTHET,GDPSI) =  (XI3(J) -XI2(J))*GENVEL(1)
     1                          *GENVEL(5)-(XI1(J) +XI2(J) -XI3(J))*
     2                          GENVEL(1)*GDTHET +(KI1(J) +XI3(J)-XI2(J))
     3                          *GENVEL(5)*GDPSI
      FIPSI(J,GDPHI,GDTHET) =  (XI2(J) -XI1(J) )*GENVEL(4)
     1                         *GENVEL(5)-(KI1(J) +XI3(J) -XI2(J))*
     2                         GENVEL(5)*GDPHI +(XI2(J) +XI3(J) -XI1(J))*
     3                         GENVEL(4)*GDTHET
      FIZJ(XMJ,XLJ,YLJ,ZLJ,GXDOTJ,GYDOTJ)=XMJ*(GENVEL(1)*(XLJ*GENVEL(4)
     1     +YLJ*GENVEL(5))-ZLJ*(GENVEL(4)**2+GENVEL(5)**2)+GENVEL(4)
     2     *(GENVEL(3)+2.0*GYDOTJ)-GENVEL(5)*(GENVEL(2)+2.0*GXDOTJ))
C
      INDEKY=INDEXY-1
      FI(1)  =  FI(1)+(XI2(M) -XI1(M) +CGMASS(M)*(XBARCM(1,M)**2
     2          -XBARCM(2,M)**2))*GENVEL(4)*GENVEL(5)-CGMASS(M)
     3          *XBARCM(2,M)*XBARCM(3,M)*GENVEL(1)*GENVEL(4)+2.0
     4          *GENVEL(1)*CGMASS(M)*XBARCM(2,M)*GENVEL(INDEXY)
     5          -GENVEL(5)*GENVEL(INDEXY+1)*(XI1(M) +XI3(M) -XI2(M))
     6          -2.0*GENVEL(4)*CGMASS(M)*XBARCM(1,M)*GENVEL(INDEXY+2)
     7          -2.0*GENVEL(5)*CGMASS(M)*XBARCM(2,M)*GENVEL(INDEXY+2)
      FI(2)  =  FI(2)-GENVEL(1)*(GENVEL(3)*CGMASS(M)+2.0*CGMASS(M)
     1          *GENVEL(INDEXY))
     2          +GENVEL(5)*(GENVEL(6)*CGMASS(M)+2.0*CGMASS(M)*
     3          GENVEL(INDEXY+2))
      FI(3)  =  FI(3)+GENVEL(1)*GENVEL(2)*CGMASS(M)
     1          -GENVEL(4)*(GENVEL(6)*CGMASS(M)+2.0*CGMASS(M)
     3          *GENVEL(INDEXY+2))
      FI(4)  =  FI(4)+CGMASS(M)*XBARCM(2,M)*XBARCM(3,M)*(GENVEL(1)**2
     1          -GENVEL(5)**2)+(XI3(M) -XI2(M) +CGMASS(M)
     2          *(XBARCM(2,M)**2-XBARCM(3,M)**2))*GENVEL(1)*GENVEL(5)
     3          +2.0*GENVEL(4)*CGMASS(M)*XBARCM(2,M)*GENVEL(INDEXY)
     4          +GENVEL(5)*GENVEL(INDEXY-1)*(XI1(M) +XI3(M) -XI2(M))
     5          +2.0*GENVEL(4)*CGMASS(M)*XBARCM(3,M)*
     6           GENVEL(INDEXY+2)
      FI(5)  =  FI(5)+(XI1(M) -XI3(M) +CGMASS(M)*(XBARCM(3,M)**2
     1          -XBARCM(1,M)**2))*GENVEL(1)*GENVEL(4)+CGMASS(M)
     2          *XBARCM(2,M)*XBARCM(3,M)*GENVEL(4)*GENVEL(5)
     3          -2.0*GENVEL(1)*CGMASS(M)*XBARCM(3,M)*GENVEL(INDEXY)
     4          -2.0*GENVEL(4)*CGMASS(M)*XBARCM(1,M)*GENVEL(INDEXY)
     5          +GENVEL(1)*GENVEL(INDEXY+1)*(XI1(M) +XI2(M) -XI3(M)) -
     6          GENVEL(4)*GENVEL(INDEXY-1)*(XI2(M) +XI3(M) -XI1(M))
     7          +2.0*GENVEL(5)*CGMASS(M)*XBARCM(3,M)
     8          *GENVEL(INDEXY+2)
      FI(6)=FI(6)+GENVEL(4)*(GENVEL(3)*CGMASS(M)+2.0*CGMASS(M)
     1          *GENVEL(INDEXY     ))-GENVEL(5)*GENVEL(2)*CGMASS(M)
      FI(KDOF)  =  FIPSI(M,GENVEL(INDEXY+1),0.0)
      KDOF  =  KDOF+1
      FI(KDOF)=FIYJ(CGMASS(M),XBARCM(1,M),XBARCM(2,M),XBARCM(3,M),0.0,
     1          GENVEL(10+2*KAXLES))
      KDOF=KDOF+1
```

SFORCE

```
      FI(KDOF) = FIPHIJ(M,0.0,GENVEL(INDEXY-1))
      KDOF=KDOF+1
      FI(KDOF)=FIZJ(CGMASS(M),XBARCM(1,M),XBARCM(2,M),XBARCM(3,M),0.0
     1          ,GENVEL(INDEXY))
      M=M+1
      KDOF = KDOF+1
      INDEXY=INDEXY+5
C
      RETURN
      END
```

SFORCE

```
      SUBROUTINE SOLVE (A,N,NDIM,NDET,IP,DET,B)
C
C     A     = ORIGINAL MATRIX.
C     N     = ACTUAL DIMENSIONS OF A.
C     NDIM  = DECLARED DIMENSION OF A IN CALLING PROGRAM.
C     NDET  = DETERMINENT CODE.
C             0 = NOT CALCULATED.
C             1 = CALCULATED.
C     IP    = INDEX OF K-TH PIVOT ROW.
C     DET   = DETERMINENT OF A.
C     B     = RIGHT HAND SIDE VECTOR.
C
C     REF. - COMMUNICATIONS OF THE ACM, APRIL 1972. PAGES 268-270,
C        AND PAGE 274.
C
      DIMENSION A(NDIM,NDIM),IP(NDIM),B(NDIM)
C
      IP(N)=1
      DO 6 K=1,N
      IF (K.EQ.N) GO TO 5
      KP1=K+1
      M=K
      DO 1 I=KP1,N
      IF (ABS(A(I,K)).GT.ABS(A(M,K))) M=I
    1 CONTINUE
      IP(K)=M
      IF (M.NE.K) IP(N)=-IP(N)
      T=A(M,K)
      A(M,K)=A(K,K)
      A(K,K)=T
      IF (T.EQ.0.0) GO TO 5
      DO 2 I=KP1,N
    2 A(I,K)=-A(I,K)/T
      DO 4 J=KP1,N
      T=A(M,J)
      A(M,J)=A(K,J)
      A(K,J)=T
      IF (T.EQ.0.0) GO TO 4
      DO 3 I=KP1,N
    3 A(I,J)=A(I,J)+A(I,K)*T
    4 CONTINUE
    5 IF (A(K,K).EQ.0.0) GO TO 15
    6 CONTINUE
      IF (NDET.EQ.0) GO TO 11
      DET=IP(N)
      DO 10 I=1,N
   10 DET=DET*A(I,I)
   11 IF (N.EQ.1) GO TO 14
      NM1=N-1
      DO 12 K=1,NM1
      KP1=K+1
      M=IP(K)
      T=B(M)
      B(M)=B(K)
      B(K)=T
      DO 12 I=KP1,N
   12 B(I)=B(I)+A(I,K)*T
      DO 13 KB=1,NM1
      KM1=N-KB
      K=KM1+1
      B(K)=B(K)/A(K,K)
      T=-B(K)
      DO 13 I=1,KM1
   13 B(I)=B(I)+A(I,K)*T
   14 B(1)=B(1)/A(1,1)
      GO TO 17
   15 WRITE (6,16)
   16 FORMAT (29H0SINGULAR MATRIX IN S/R SOLVE)
      N = 0
   17 RETURN
      END
```

SOLVE

```
      SUBROUTINE SUMARY
C
C     OUTPUT MINIMUM-MAXIMUM INFORMATION
C
      COMMON/CALC/    BETA (3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1   FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2   PITCH,PTRIAL(5),ROLL,TIME,TMIN,KBARCM(3,13),YAW
      COMMON/MAXMIN/ AMAX(50),AMIN(50),ANGLE(19),CIP(19,6),
     1   CIQ(19,6),CPSIQ(19,5),CPSO(19,6),CRKM(19,5),DMAX(50),
     2   DMIN(50),IFL(19,6),JMAX,KMAX,SMAX(100),SMIN(100),
     3   TAMAX(50),TAMIN(50),TDMAX(50),TDMIN(50),TSMAX(100),
     4   TSMIN(100),TVMAX(50),TVMIN(50),VMAX(50),VMIN(50),
     5   YIELD(6)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1   KGWIR2,KFACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2   NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3   XBOGIE(2),IDD(40)
C
      WRITE (6,1)
C
C     DOF
C
      WRITE (6,2)
      WRITE (6,5)
      DO 100 INDEX=1,MDOF
  100 WRITE (6,3)  INDEX,DMAX(INDEX),TDMAX(INDEX),DMIN(INDEX),
     1             TDMIN(INDEX)
C
      WRITE (6,4)
      WRITE (6,5)
      DO 200 INDEX=1,MDOF
  200 WRITE (6,3)  INDEX,VMAX(INDEX),TVMAX(INDEX),VMIN(INDEX),
     1             TVMIN(INDEX)
C
      WRITE (6,6)
      WRITE (6,5)
      DO 300 INDEX=1,MDOF
  300 WRITE (6,3)  INDEX,AMAX(INDEX),TAMAX(INDEX),AMIN(INDEX),
     1             TAMIN(INDEX)
      IF (MASSES.EQ.1) GO TO 500
C
      WRITE (6,7)
      DO 400 INDEX=1,MSPRNG
  400 WRITE (6,8)  INDEX,SMAX(INDEX),TSMAX(INDEX),SMIN(INDEX),
     1             TSMIN(INDEX)
C
  500 CRIT(1) = DDMIN/DCG0
      IF (DVEL.GT.0.) WRITE (6,9) DVEL,TMIN
      IF (DVEL.EQ.0.) WRITE (6,10) DCG0,DDMIN,TMIN,CRIT(1)
      IF (TMIN.EQ.TIME.AND.DVEL.EQ.0.0) WRITE (6,11)
      RETURN
C
C
C     FORMAT STATEMENTS.
C
    1 FORMAT (//20H SUMMARY INFORMATION//)
    2 FORMAT (14H DISPLACEMENTS,/)
    5 FORMAT (5H DOF ,5(1H.),7HMAXIMUM,7(1H.),
     1       4HTIME,2X,6(1H.),7HMINIMUM,8(1H.),4HTIME,/)
    3 FORMAT (I4,2E13.5,1X,2E13.5)
    4 FORMAT (11H0VELOCITIES,/)
    6 FORMAT (14H0ACCELERATIONS,/)
    7 FORMAT (22H0SPRING MAX DEFLECTION,7(1H.), 4HTIME,4(1H.),
     1   14HMIN DEFLECTION,5(1H.),4HTIME,/)
    8 FORMAT (I7,2X,2E13.5,3X,2E13.5)
    9 FORMAT (19H0VEHICLE OVERTURNED/
     1   14H    VELOCITY = E15.6/14H    TIME       = E15.5)
   10 FORMAT (25H0VEHICLE REMAINED UPRIGHT/
     1   25H    INITIAL CG DISTANCE = E15.6/
     2   25H    MINIMUM CG DISTANCE = E15.6/
     3   25H    TIME OF MINIMUM     = E15.6/
     4   25H    CRIT                = E15.6)
   11 FORMAT (78H0*** WARNING - MINIMUM CG DISTANCE OCCURRED AT LAST INT
```

SUMARY

```
1EGRATION TIME STEP  ***)
 END
```

SUMARY

```
      SUBROUTINE SUMTAB (N,KJK)
      COMMON/BLAST/   A,AD,AD1,AD2,ALT,A0,COSA,DELTA,KBLAST,PIMP,PR,
     1    PS,PSOEST,PS0,P0,QIMP,Q0,RANGE,RANGEM,SF(5),SINA,VS,W,Z
      COMMON/CALC/     BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/MAXMIN/ AMAX(50),AMIN(50),ANGLE(19),CIP(19,6),
     1    CIQ(19,6),CPSIQ(19,6),CPS0(19,6),CRKM(19,6),DMAX(50),
     2    DMIN(50),IFL(19,6),JMAX,KMAX,SMAX(100),SMIN(100),
     3    TAMAX(50),TAMIN(50),TDMAX(50),TDMIN(50),TSMAX(100),
     4    TSMIN(100),TVMAX(50),TVMIN(50),VMAX(50),VMIN(50),
     5    YIELD(6)
      COMMON/OPTION/ IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IDD(40)
      DIMENSION KFLAG(2)
      DATA KFLAG/2H   ,2H* /
C
      IF (N-1) 50,200,800
C     ONLY ENOUGH TABLE SPACE FOR 6 YIELDS AND 19 ANGLES.
   50 JMAX = 0
      KMAX = 0
      IF (KDAM.EQ.0) GO TO 150
      DO 100 K=1,6
      DO 100 J=1,19
      CPS0(J,K) = 0.
      CIQ(J,K) = 0.
      CPSIQ(J,K) = 0.
      CIP(J,K) = 0.
      CRKM(J,K) = 0.
  100 IFL(J,K) = KFLAG(1)
  150 RETURN
C
  200 IF (JMAX.GT.0) GO TO 300
      JMAX = 1
      KMAX = 1
      ANGLE(1) = AD
      YIELD(1) = W
      J = 1
      K = 1
      RETURN
C
  300 DO 400 JJ=1,JMAX
      J = JJ
      IF (ANGLE(JJ).EQ.AD) GO TO 500
  400 CONTINUE
      JMAX = JMAX + 1
      J = JMAX
      ANGLE(JMAX) = AD
  500 DO 600 KK = 1,KMAX
      K = KK
      IF (YIELD(KK).EQ.W) GO TO 700
  600 CONTINUE
      KMAX = KMAX + 1
      K = KMAX
      YIELD(KMAX) = W
  700 RETURN
C
  800 IF (JMAX.EQ.0) GO TO 900
      IF (KERR.GT.0) IFL(J,K) = KFLAG(2)
      IF (KJK.NE.1) IFL(J,K) = KFLAG(2)
      IF (TMIN.EQ.TIME.AND.DVEL.EQ.0.0) IFL(J,K) = KFLAG(2)
      CPS0(J,K) = PS0
      CIQ(J,K) = QIMP
      CPSIQ(J,K) = PS0*QIMP
      CRKM(J,K) = RANGEM/1000.
      CIP(J,K) = PIMP
  900 RETURN
      END
```

SUMTAB

```
      SUBROUTINE TIRES(XSC,GF)
C
C     SUBROUTINE TIRES FINDS THE GENERALIZED TIRE FORCE
C
      COMMON/ADAM/    KSTART,NEQ,NIT,QR(100),QRD(6,100),QRP(100),TIO12,
     1    TIO2,TIO24,TIO72
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FAERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1    DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2    XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTX1,DPRT,DPRT1,ENDTX
      COMMON/INTFLG/  IFIRST
      COMMON/JFTRIM/  JTRIM
      COMMON/MOVING/  XDSAVE,XOBAR,XSAVE,XTB(5),YDSAVE,YOBAR,YSAVE,
     1    ZDSAVE,ZOBAR,ZSAVE,ZTB(5)
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IDD(40)
      COMMON/SPRING/  FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1    X(600)
      COMMON /TIREC/  CN,CT,NTIRES(6),NTSPRN,RW,TLX(3,6),TLY(6),TLZ(6),U
      COMMON /TIRSAV/ DELTNX(3,2,18),DELTN1(3,2,18),DTTNX(2,18),
     1  DTTN1(2,18),SLID(2,18)
C
      DIMENSION GF(1)
      DIMENSION BATA(3,3)
      DIMENSION A(2),AM(3),BATAT(3,3),CK(2),DELTD(3),DIF(3),E(3),EU(3),
     1FORB(3),GMAT(3,2),GMSQ(3,3),GVN(3),GVT(3),HNK(2),POSD(3),RNK(3),
     2TGS(2),VB(3),XTBAR(2),YTBAR(2),ZTBAR(2)
      DIMENSION DD(3),GVTD(3),EUF(3)
      EQUIVALENCE  (BETA(1,1),BATA(1,1))
C
C     TRANSPOSE BETA MATRIX.
      DO 20 ID=1,3
      DO 20 IE = 1,3
   20 BATAT(ID,IE) = BATA(IE,ID)
      DO 50 ID=1,MDOF
   50 GF(ID) = 0.0
      IF (KTIRE.GT.0) GO TO 80
C
      KTIRE = 1
      IT = 0
      DO 55 IA = 1,KAXLES
      NTT = NTIRES(IA)
      DO 55 NT = 1,NTT
      IT = IT + 1
      DO 55 IROL=1,2
      SLID(IROL,IT) = 0.
      DTTN1(IROL,IT) = 0.0
      DO 55 ID = 1,3
   55 DELTN1(ID,IROL,IT) = 0.
      DO 57 ID=1,3
      GVN(ID) = 0.
   57 GVT(ID) = 0.
      GVNT = 0.
      GVTT = 0.
      VT = 1.0
      EPS = 0.0
      NX=NPPC(NTSPRN-1)
      NX1=NPPC(NTSPRN)
      JP=JCURVE(NTSPRN-1)
      JP1=JCURVE(NTSPRN)
      NDOFA = 6 +2*KAXLES
      IF (KAXLES.EQ.1) NDOFA=3
      TGS(1) = TAN(GS(1))
      TGS(2) = TAN(GS(2))
      A(1) = 0.0
      A(2) = 0.0
      IF(XSC.GE.0.0) A(2) = -XSC*(TGS(2) -TGS(1))
      IF(XSC.LT.0.0) A(1) = XSC*(TGS(2) -TGS(1))
      SE = SIN(EPS)
```

```
      CE = COS(EPS)
      DO 60 IK=1,2
      SGS = SIN(GS(IK))
      CGS = COS(GS(IK))
      GMAT(1,IK) = -CE*SGS
      GMAT(2,IK) = CGS
      GMAT(3,IK) = SE*SGS
   60 CONTINUE
      NK =2
      IF(GS(1).EQ.GS(2)) NK=1
C
   80 IT = 0
      TLYB = 0.
      DO 1000 IA = 1,KAXLES
      IQRA = 5 +2*IA
      IQYA = IQRA +1
      IF (MASSES.GT.1) TLYB = TLY(IA) - XBARCM(2,IA+1)
      NTT=NTIRES(IA)
      DO 1000 NT=1,NTT
      IT = IT + 1
      TLXB = TLX(NT,IA)
C     CALCULATE CENTER WHEEL POSITION
      DO 100 IROL=1,2
      IF(IROL.EQ.2) TLXB=-TLXB
      DISX=TLXB-TLYB*GENDIS(IQRA)
      DISY=GENDIS(IQYA)+TLY(IA)+TLXB*GENDIS(IQRA)
      XTBAR(IROL) = XOBAR +BATA(1,1)*DISX +BATA(1,2)*DISY+
     1BATA(1,3)*TLZ(IA)
      YTBAR(IROL) = YOBAR +BATA(2,1)*DISX +BATA(2,2)*DISY +
     1BATA(2,3)*TLZ(IA)+RW-TLY(IA)
      ZTBAR(IROL) = ZOBAR +BATA(3,1)*DISX +BATA(3,2)*DISY
     1+BATA(3,3)*TLZ(IA)
  100 CONTINUE
      IF (IA.NE.1) GO TO 120
      IF (NT.NE.NTT) GO TO 140
      DX = -BATA(1,2)*RW
      DZ = -BATA(3,2)*RW
      XTB(1)=XTBAR(1) + DX
      XTB(2)=XTBAR(2) + DX
      ZTB(1)=ZTBAR(1) + DZ
      ZTB(2)=ZTBAR(2) + DZ
      XTB(5)=XTB(1)
      ZTB(5)=ZTB(1)
      GO TO 140
  120 IF (IA.NE.KAXLES) GO TO 140
      IF (NT.NE.NTT) GO TO 140
      XTB(3)=XTBAR(2) + DX
      XTB(4)=XTBAR(1) + DX
      ZTB(3)=ZTBAR(2) + DZ
      ZTB(4)=ZTBAR(1) + DZ
  140 CONTINUE
C     CALCULATE HEIGHT OF WHEEL ABOVE GROUND
      DIF(1) = XTBAR(2) -XTBAR(1)
      DIF(2) = YTBAR(2) -YTBAR(1)
      DIF(3) = ZTBAR(2) -ZTBAR(1)
      AL = SQRT(DIF(1)**2 + DIF(2)**2 + DIF(3)**2)
      XI = DIF(1)*CE -DIF(3)*SE
      TLXB = TLX(NT,IA)
      DO 1000 IROL=1,2
      IF (IROL.EQ.2) TLXB = -TLXB
      FTD = 0.0
      KOFK =1
      DO 200 IK=1,NK
      CK(IK) = AL/SQRT(AL**2*(1.0 +TGS(IK)**2) -(DIF(2)
     1-TGS(IK)*XI)**2)
      HWK(IK) = (YTBAR(IROL) -TGS(IK)*(XTBAR(IROL)*CE
     1-ZTBAR(IROL)*SE) -A(IK))*CK(IK)
  200 CONTINUE
      HW = HWK(1)
      IF(NK.EQ.1) GO TO 225
      IF (GS(2).GT.GS(1)) GO TO 210
      IF (HWK(1).GE.HWK(2)) GO TO 225
      GO TO 220
```

```
  210 IF (HWK(1).LE.HWK(2)) GO TO 225
  220 HW = HWK(2)
      KOFK =2
C     CALCULATE TIRE NORMAL DEFLECTION
  225 DELN = HW -RW
C
      IF(DELN.LE.0.0) GO TO 275
      SLID(IROL,IT) = -1.
      DTTN1(IROL,IT) = 0.
      DO 250 ID=1,3
      DELTN1(ID,IROL,IT) = 0.
  250 CONTINUE
      GO TO 1000
  275 CONTINUE
C     CALCULATE BODY-AXIS COMPONENTS OF UNIT GROUND
C     SURFACE NORMAL VECTOR
      CALL MULT (BATAT,GMAT(1,KOFK),RNK,3,3,3,3,1)
      IF (TIME.GE.0.0) GO TO 300
      IF (IOPT.EQ.1.OR.IA.GT.1) GO TO 400
      IF (MASSES.EQ.1) GO TO 400
      GO TO 325
C     CALCULATE WHEEL VELOCITY COMPONENTS IN BODY AXES.
  300 VB(1) = -(TLY(IA)-RW)*GENVEL(1) + GENVEL(2) + TLZ(IA)*GENVEL(5) -
     1   (TLYB-RW)*GENVEL(IQRA)
      VB(2) =TLXB*GENVEL(1)+GENVEL(3)-TLZ(IA)*GENVEL(4)+TLXB*GENVEL(IQRA)
     1+GENVEL(IQYA)
      VB(3) = (TLY(IA)-RW)*GENVEL(4) - TLXB*GENVEL(5) + GENVEL(6)
C
C     CALCULATE WHEEL VELOCITY NORMAL TO GROUND AND BODY AXIS
C     COMPONENTS THERE OF
C
      GVNT = RNK(1)*VB(1) + RNK(2)*VB(2) + RNK(3)*VB(3)
      DO 310 IJ=1,3
      GVN(IJ) = RNK(IJ)*GVNT
  310 CONTINUE
C     CALCULATE BODY-AXIS COMPONENTS OF WHEEL VELOCITY PARALLEL
C     TO GROUND AND TOTAL VELOCITY
      DO 320 IJ=1,3
  320 GVT(IJ) = VB(IJ) - GVN(IJ)
      IF (IOPT.EQ.1.OR.IA.GT.1) GO TO 360
      IF (MASSES.EQ.1) GO TO 360
C
C     FOR FRONT AXLE, TRUCK ONLY.
C
C     CALCULATE COMPONENTS OF UNIT VECTOR IN GROUND PLANE
C     NORMAL TO INTERSECTION OF WHEEL PLANE AND GROUND PLANE.
C     CALCULATE COMPONENTS OF VECTOR IN AXLE DIRECTION
  325 CALL MULT(BATAT,DIF,AM,3,3,3,3,1)
C     CALCULATE THE DOT PRODUCT OF THE AXLE VECTOR AND THE
C     SURFACE NORMAL VECTOR
      AN = AM(1)*RNK(1) + AM(2)*RNK(2) + AM(3)*RNK(3)
C
C     CALCULATE THE COMPONENTS OF THE VECTOR IN DESIRED
C     DIRECTION             AND MAGNITUDE OF E VECTOR
      EMAG = 0.0
      DO 330 II=1,3
      E(II) = RNK(II)*AN -AM(II)
      EMAG = EMAG +E(II)**2
  330 CONTINUE
      EMAG = SQRT(EMAG)
C
C     CALCULATE DESIRED UNIT VECTOR, BODY AXES(EU) AND FIXED AXES(EUF).
C
      DO 340 II=1,3
      EU(II) = E(II)/EMAG
  340 CONTINUE
      DO 345 II = 1,3
  345 EUF(II) = BATA(II,1)*EU(1) + BATA(II,2)*EU(2) + BATA(II,3)*EU(3)
      IF (TIME.LT.0.0) GO TO 400
C
C     CALCULATE THE MAGNITUDE OF COMPONENTS OF THE WHEEL
C     VELOCITY PARALLEL TO GROUND IN DIRECTION OF E VECTOR
      GVTT = GVT(1)*EU(1) + GVT(2)*EU(2) + GVT(3)*EU(3)
```

TIRES

```
C       CALCULATE BODY AXIS COMPONENTS OF GVT
        DO 350 II=1,3
  350   GVT(II) = EU(II) * GVTT
        GVTT=ABS(GVTT)
        GO TO 365
C
C       FOR ALL OTHER AXLES.
C
  360   GVTT = GVT(1)**2 + GVT(2)**2 + GVT(3)**2
        GVTT = SQRT(GVTT)
  365   VT = GVTT
        IF (VT.EQ.0.0) VT = 1.0
        D = DTTN1(IROL,IT)
        IF (D.EQ.0.0) D = 1.0
        DO 370 ID=1,3
        DD(ID) = DELTN1(ID,IROL,IT)/D
  370   GVTD(ID) = (BATA(ID,1)*GVT(1) + BATA(ID,2)*GVT(2) +
       1  BATA(ID,3)*GVT(3))/VT
C       FIND NORMAL SPRING FORCE FROM DEFLECTION DELN
  400   CONTINUE
        CALL INT1(DELN,NX,X(JP),FOFX(JP),FN)
        FN = FN/CK(KOFK)
C       FIND TANGENTIAL SPRING FORCE ON THE TIRE
C       CALCULATE MAXIMUM TANGENTIAL SPRING FORCE
C
        FTMAX = U*FN
        IF(TIME.LT.0.0) GO TO 450
        DEL = 0.
        VDT = GVTT*DELTIM
        IF (SLID(IROL,IT)) 405,410,600
  405   SLID(IROL,IT) = 0.
        FT = 0.
        DELTD(1) = 0.
        DELTD(2) = 0.
        DELTD(3) = 0.
        DELTT = 0.
        GO TO 700
C       CALCULATE NEW DISPLACEMENT COMPONENTS.
  410   DO 425 ID=1,3
  425   DELTD(ID) = (DTTN1(IROL,IT) - DEL)*DD(ID) + VDT*GVTD(ID)
        GO TO 470
C       FOR TRIM ONLY
  450   CALL MULT (GMAT(1,KOFK),GMAT(1,KOFK),GMSQ,3,1,3,1,3)
        DO 460 ID=1,3
        DO 460 IE=1,3
        GMSQ(IE,ID) = -GMSQ(IE,ID)
        IF(IE.EQ.ID) GMSQ(IE,ID) = 1.0 +GMSQ(IE,ID)
  460   CONTINUE
        POSD(1)=XTBAR(IROL)-TLKB
        POSD(2) = YTBAR(IROL) -RW
        POSD(3)=ZTBAR(IROL)-TLZ(IA)
        CALL MULT (GMSQ,POSD,DELTD,3,3,3,3,1)
  470   IF (IOPT.EQ.1.OR.IA.GT.1) GO TO 475
        IF (MASSES.EQ.1) GO TO 475
C
C       FRONT AXLE, TRUCK ONLY.
C
        DELTT = DELTD(1)*EUF(1) + DELTD(2)*EUF(2) + DELTD(3)*EUF(3)
        DO 472 II=1,3
  472   DELTD(II) = EUF(II)*DELTT
        DELTT=ABS(DELTT)
        GO TO 480
C       FOR ALL OTHER AXLES.
C       CALCULATE DELT
  475   DELTT = DELTD(1)**2 + DELTD(2)**2 + DELTD(3)**2
        DELTT = SQRT(DELTT)
  480   IF (SLID(IROL,IT).EQ.1.0) GO TO 700
C       CALCULATE SPRING FORCE FT
        CALL INT1 (DELTT,NX1,X(JP1),FOFX(JP1),FT)
        IF (JTRIM.EQ.0) GO TO 700
C       IF FT LE TO FT MAX ACCEPT FT
        IF(FT.LE.FTMAX) GO TO 700
        IF (TIME.GE.0.0) GO TO 485
```

```
      WRITE (6,1)FT,DELTT,FTMAX,U,FN
      KERR = 2
      RETURN
C     START SLIDING.
  485 SLID(IROL,IT) = 1.
      FT = FTMAX
      CALL INT1 (FT,NX1,FOFX(JP1),X(JP1),DELTM)
      IF (DELTT.EQ.0.) DELTT = 1.
      DO 490 ID = 1,3
  490 DELTD(ID) = DELTM*DELTD(ID)/DELTT
      DELTT = DELTM
      GO TO 700
C     TIRE IS SLIDING.
  600 DV = DD(1)*GVTD(1) + DD(2)*GVTD(2) + DD(3)*GVTD(3)
      CALL INT1 (FTMAX,NX1,FOFX(JP1),X(JP1),DELTM)
      FT = FTMAX
      IF (DELTM.GT.VDT) GO TO 650
      DO 620 ID=1,3
  620 DELTD(ID) = VDT*GVTD(ID)
      GO TO 470
  650 DEL = DTTN1(IROL,IT ) + DV*VDT - SQRT(DELTM**2 - (1.-DV**2)*VDT**2)
C     CHECK TO SEE IF SLIDING IS CONTINUING.
      IF (DEL.GT.0.0) GO TO 410
C     SLIDING IS STOPPING.
      SLID(IROL,IT) = 0.
      DEL = 0.
      GO TO 410
  700 CONTINUE
C     CALCULATE NORMAL DAMPING FORCES
C
      FND=CN*GVNT
C
C     CALCULATE TANGENTIAL DAMPING FORCE
      IF (SLID(IROL,IT).EQ.0..AND.FT.LT.FTMAX) FTD = CT*GVTT
C     FIND TOTAL FORCES IN BODY AXIS COMPONENTS
C
      TERM1=FN-FND
      DO 750 ID=1,3
  750 FORB(ID) = TERM1*RNK(ID) - FTD*GVT(ID)/VT
      IF (DELTT.EQ.0.) GO TO 800
      DO 760 ID = 1,3
  760 DD(ID) = DELTD(ID)/DELTT
      CALL MULT (BATAT,DD,EU,3,3,3,3,1)
      DO 775 ID=1,3
  775 FORB(ID) = FORB(ID) - FT*EU(ID)
C     CALCULATE THE GENERALIZED FORCE.
  800 GF(1)=GF(1)-FORB(1)*(TLY(IA)-RW)+FORB(2)*TLXB
      GF(2) = GF(2) +FORB(1)
      GF(3) = GF(3) +FORB(2)
      IF (MASSES.EQ.1) GO TO 855
      GF(4)=GF(4)-FORB(2)*TLZ(IA)+FORB(3)*(TLY(IA)-RW)
      GF(5)=GF(5)+FORB(1)*TLZ(IA)-FORB(3)*TLXB
      GF(6) = GF(6) +FORB(3)
      GF(IQRA)=GF(IQRA)+FORB(2)*TLXB
      IF (IOPT .NE.1) GF(IQRA) = GF(IQRA) - FORB(1)*
     1(TLYB - RW)
      GF(IQYA) = GF(IQYA) +FORB(2)
C     SET UP N-1 AND N-2 FOR NEXT TIME STEP
  855 IF (TIME.LT.0.0) GO TO 1000
  860 DO 900 ID=1,3
  900 DELTN1(ID,IROL,IT) = DELTD(ID)
      DTTN1(IROL,IT) = DELTT
 1000 CONTINUE
      RETURN
C
    1 FORMAT (10X,30HTRUCK SLIDING IN TRIM. (TIRES)/,
     115X,22HFORCE (TESTED)          =,
     1E13.5,/,16X,21HDEFLECTION (CALC)    =,E13.5,//,15K,
     222HMAX. FORCE              =,E13.5,/,16X,
     321HMU (INPUT)             =,E13.5,/,16X,21HNORMAL FORCE (CALC) =,
     4E13.5)
      END
```

TIRES

```
      SUBROUTINE TRIM
C
C
C     SUBROUTINE TRIM TRIMS THE SYSTEM.
C
      COMMON/BOTM/    EXPAN(100),NBOTOM(100),SBOTM(100),VEL(100)
      COMMON/CALC/    BETA(3,3),CRIT(5),DCG,DCG0,DDMIN,DVEL,
     1    FNERO(50),FORCE(50),GENACC(50),GENDIS(50),GENVEL(50),KTIRE,
     2    PITCH,PTRIAL(5),ROLL,TIME,TMIN,XBARCM(3,13),YAW
      COMMON/CHECK/   JDEBUG
      COMMON/DATAIN/  CGMASS(13),CGPOS(3,13),DAMPF(600),DAMPV(600),
     1    DELTIM,ENDTIM,G,GS(2),ISDAMP(50),MDIMDP,NPDAMP(50),SLOPE,
     2    XIYZ,XI1(13),XI2(13),XI3(13),DELTX,DELTK1,DPRT,DPRT1,ENDTX
      COMMON/DELTAS/  DISP(100)
      COMMON/JFTRIM/  JTRIM
      COMMON XLAMDA(100,44)
      COMMON/MOVING/  XOSAVE,XOBAR,XSAVE,XTB(5),YDSAVE,YOBAR,YSAVE,
     1    ZDSAVE,ZOBAR,ZSAVE,ZTB(5)
      COMMON/OPTION/  IOPT,JGRAPH,KAXLB(2),KAXLES,KBOG,KDAM,KERR,KGWIR1,
     1    KGWIR2,KRACKS,KRIGID,KSHELT,KWIRES,MASSES,MDOF,MSPRNG,
     2    NASPR(6),NCALL,NCASE,NJOB,NOUT,NPRINT,NRSPR(3),NSPRNG,NTRIAL,
     3    XBOGIE(2),IDD(40)
      COMMON/SPRING/  FOFX(600),JCURVE(52),MAXSPR,NPPC(52),
     1    X(600)
      COMMON /TIREC/  CN,CT,NTIRES(6),NTSPRN,RW,TLX(3,6),TLY(6),TLZ(6),U
      COMMON/TRMDIM/  ER(50)
      COMMON/WEIGHT/  WEIGHT,WGTS(13)
C
C
      KTIRE = 0
      JTRIM=0
      KDOF=MDOF
      IF (KAXLES.EQ.1) KDOF = 8
      DO 10 INDEX=1,KDOF
      GENDIS(INDEX)=0.0
      GENVEL(INDEX) = 0.
   10 ER(INDEX)=0.001
      IF(KWIRES.GT.0) CALL PRETRM
      IF (KERR.GT.0) RETURN
      N=0
      DO  15 INDEX=1,KAXLES
   15 N=NTIRES(INDEX)+N
      TOTAL=2*N
      ISTART=JCURVE(NTSPRN-1)
      NP=NPPC(NTSPRN-1)
      CALL INT1(-.001,NP,X(ISTART),FOFX(ISTART),F1)
      CALL INT1(0.0,NP,X(ISTART),FOFX(ISTART),F2)
      TS=1000.0*(F2-F1)*TOTAL
      GENDIS(3)=WEIGHT/TS
C
      NPR = 0
      IF (JDEBUG.EQ.2) NPR = 1
      NCOUNT=0
      NTRY=0
      NOK=0
      ROLL=GS(1)
      IF (SLOPE.LT.0.0) ROLL=GS(2)
      GENDIS(1)=ROLL
      GENDIS(2)=TLY(1)*ROLL
      PITCH=0.0
      YAW=0.0
   20 CALL BETAIJ
      XOBAR=0.5*((1.0+BETA(1,1))*GENDIS(2)+BETA(1,2)*GENDIS(3)
     1        +BETA(1,3)*GENDIS(6))
      YOBAR=0.5*(BETA(2,1)*GENDIS(2)+(1.0+BETA(2,2))*GENDIS(3)
     1        +BETA(2,3)*GENDIS(6))
      ZOBAR=0.5*(BETA(3,1)*GENDIS(2)+BETA(3,2)*GENDIS(3)+(1.0+BETA(3,3))
     1        *GENDIS(6))
C
C     COMPUTE FORCES.
C
      CALL EXTRNL
      IF (JTRIM.EQ.1) GO TO 25
```

```
      IF (JDEBUG.EQ.2) WRITE (6,9000) (FORCE(IO),IO=1,MDOF)
      CALL RELAX (MDOF,FORCE,GENDIS,ER,NOK,NPR,NCOJNT)
   25 ROLL=GENDIS(1)
      PITCH=GENDIS(4)
      YAW=GENDIS(5)
C
C     HAVE WE TRIMMED THE VEHICLE ?
C
      IF (NCOUNT.EQ.1) NTRY=NTRY+1
      IF (NTRY.GT.20) GO TO 1000
      IF (NOK.EQ.0) GO TO 20
      IF (NOK.EQ.2) GO TO 1000
C
C     VEHICLE IS TRIMMED.
C
C
C     OUTPUT TRIM CONDITIONS.
C
      JTRIM=JTRIM+1
      IF (JTRIM.LT.2) GO TO 20
      WRITE (6,4)
      IF(JDEBUG.GT.0) WRITE (6,3) (GENDIS(IO),IO=1,MDOF)
C
C     SET UP FOR INTEGRATION.
C
      XSAVE=XOBAR
      YSAVE=YOBAR
      ZSAVE=ZOBAR
C
      RETURN
C
C     CAN'T FIND A SOLUTION.
C
 1000 WRITE (6,1)
      KERR = 2
      RETURN
C
    1 FORMAT (33H0**** VEHICLE TRIM ABORTED (TRIM))
    3 FORMAT (31H0GENERALIZED TRIM COORDINATES =,
     2       /,(1X,10E13.5))
    4 FORMAT (21H VEHICLE HAS TRIMMED.)
 9000 FORMAT (14H TRIM FORCES =,/,(1X,10E12.4))
C
      END
```

TRIM

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314
3	Director Defense Advanced Research Projects Agency ATTN: Tech Lib NMRO PMO 1400 Wilson Boulevard Arlington, VA 22209
4	Director of Defense Research and Engineering ATTN: DD/TWP DD/S&SS DD/I&SS AD/SW Washington, DC 20301
1	Director Weapons Systems Evaluation Gp ATTN: Document Control Washington, DC 20305
1	Director Institute for Defense Analyses ATTN: IDA Librarian, Ruth S. Smith 400 Army-Navy Drive Arlington, VA 22202
2	Asst. to the Secretary of Defense (Atomic Energy) ATTN: Document Control Donald R. Cotter Washington, DC 20301
5	Director Defense Intelligence Agency ATTN: DT-IC DT-7D/E. O. Farrell DT-2/Wpns & Sys Div Technical Library DI-7E Washington, DC 20301
6	Director Defense Nuclear Agency ATTN: SPTD/Mr. J. Kelso STSI/Archives SPAS/Mr. J. Moulton STSP STVL/Dr. La Vier RATN/Cdr Alderson Washington, DC 20305
6	Director Defense Nuclear Agency ATTN: DDST/Mr. P. Haas DDST/Mr. M. Atkins STTL/Tech Lib (2 cys) SPSS (2 cys) Washington, DC 20305
2	Commander Field Command, DNA ATTN: FCPR FCTMOF Kirtland AFB, NM 87115
1	Chief Las Vegas Liaison Office Field Command TD, DNA ATTN: Document Control P. O. Box 2702 Las Vegas, NV 89104
1	Commander Field Command, DNA Livermore Branch ATTN: FCPRL P. O. Box 808 Livermore, CA 94550

DISTRIBUTION LIST

No. of Copies	Organization
1	Director Defense Communications Agency ATTN: Code 930 Washington, DC 20305
3	Director Joint Strategic Target Planning Staff JCS ATTN: Sci & Tech Info Lib JLTW-2 DOXT Offut AFB Omaha, NB 68113
1	Director National Security Agency ATTN: E. F. Butala, R15 Ft. George G. Meade, MD 20755
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: Technical Library 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: W. H. Hubbard 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Aviation Research and Development Command ATTN: DRSAV-E 12th and Spruce Streets St. Louis, MO 63166

No. of Copies	Organization
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
5	Commander US Army Electronics Command ATTN: DRSEL-RD DRSEL-TL-IR R. Freiberg J. Roma A. Sigismondi E. T. Hunter Fort Monmouth, NJ 07703
3	Commander US Army Missile Materiel and Readiness Command ATTN: DRSMI-R DRSMI-XS/Chief Scientist Technical Library Redstone Arsenal, AL 35809
1	Commander US Army Missile Research and Development Command ATTN: DRDMI-R Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Research & Development Cmd ATTN: DRDTA-RWL Warren, MI 48090
1	Commander US Army Mobility Equipment Research & Development Cmd ATTN: DRXFB-RT, Dr.Kenneth Oscar Fort Belvoir, VA 22060

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
5	Commander US Army Armament Research and Development Command ATTN: DRDAR-TSS (2 cys) DRDAR-LCN-F Mr. Warren Reiner Mr. P. Angelotti (2 cys) Dover, NJ 07801
1	Commander US Army Watervliet Arsenal Watervliet, NY 12189
5	Commander US Army Harry Diamond Labs ATTN: DRXDO-TI DRXDO-TI/012 DRXDO-NP DRXDO-RBH Mr. P.A. Caldwell DELHD-RBA, J. Rosado 2800 Powder Mill Road Adelphi, MD 20783
5	Commander US Army Harry Diamond Labs ATTN: Mr. James Gaul Mr. L. Belliveau Mr. J. Gwaltney Mr. F. N. Wimenitz Mr. Bill Vault 2800 Powder Mill Road Adelphi, MD 20783
3	Commander US Army Materials and Mechanics Research Center ATTN: Technical Library John Mescall Richard Shea Watertown, MA 02172
2	Commander US Army Natick Research and Development Command ATTN: DRXRE, Dr. D. Seiling DRXNM-UE Arthur Johnson Natick, MA 01762
1	Commander US Army Foreign Science and Technology Center ATTN: Rsch & Concepts Branch 220 7th Street, NE Charlottesville, VA 22901
2	Commander US Army Nuclear Agency ATTN: ACTA-NAW Tech Library 7500 Backlick Rd., Bldg 2073 Springfield, VA 22150
1	Commander US Army Training and Doctrine Command ATTN: ATCD-SA, Mr. Oscar Wells Fort Monroe, VA 23651
2	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib LTC John Hesse White Sands Missile Range NM 88002
1	Commander Combined Arms Combat Developments Activity ATTN: ATCA-CO, Mr. L.C. Pleger Fort Leavenworth, KS 66027
1	Commander US Army Logistics Center ATTN: ATCL-SCA, Robert Cameron Fort Lee, VA 23801

DISTRIBUTION LIST

No. of Copies	Organization
1	Interservice Nuclear Weapons School ATTN: Technical Library Kirtland AFB, NM 87115
2	Director Defense Civil Preparedness Agency ATTN: Mr. George Sisson/RF-SR Technical Library Washington, DC 20301
2	HQDA (DAMA-AR; NCL Div) Washington, DC 20310
2	Deputy Chief of Staff for Operations and Plans ATTN: Technical Library Director of Chemical and Nuclear Operations Department of the Army Washington, DC 20310
1	Commander US Army Research Office P. O. Box 12211 Research Triangle Park NC 27709
1	Program Manager US Army BMD Program Office ATTN: John Shea 5001 Eisenhower Avenue Alexandria, VA 22333
2	Director US Army BMD Advanced Technology Center ATTN: CRDABH-X CRDABH-S Huntsville, AL 35807
1	Commander US Army BMD Systems Command ATTN: BDMSC-TFN/N.J. Hurst P. O. Box 1500 Huntsville, AL 35807
2	Office, Chief of Engineers Department of the Army ATTN: DAEN-MCE-D DAEN-RDM 890 South Pickett Street Alexandria, VA 22304
1	Commander US Army Engineering Center ATTN: ATSEN-SY-L Fort Belvoir, VA 22060
1	Division Engineer US Army Engineering Division Ohio River ATTN: Docu Cen P. O. Box 1159 Cincinnati, OH 45201
1	Division Engineer US Army Engineering Division ATTN: HNDSE-R/M.M. Dembo Huntsville Box 1600 Huntsville, AL 35804
5	Commander US Army Engineer Waterways Experiment Station ATTN: Tech Library William Flathau John N. Strange Guy Jackson Leo Ingram P. O. Box 631 Vicksburg, MS 39180

DISTRIBUTION LIST

No. of Copies	Organization
3	Office of Naval Research ATTN: Code 461 Jacob L. Warner Thomas P. Quinn N. Perrone 800 N. Quincy Street Arlington, VA 22217
2	Chief of Naval Operations ATTN: OP-03EG OP-985F Department of the Navy Washington, DC 20350
1	Chief of Naval Material ATTN: MAT 0323 Department of the Navy Arlington, VA 22217
3	Director Strategic Systems Projects Ofc ATTN: NSP-43, Tech Lib NSP-273 NSP-272 Department of the Navy Washington, DC 20360
1	Commander Naval Electronic Systems Cmd ATTN: PME 117-21A Washington, DC 20360
2	Commander Naval Sea Systems Command ATTN: ORD-91313 Library Code 03511 Department of the Navy Washington, DC 20362
3	Commander Naval Facilities Engineering Cmd ATTN: Code 03A Code 04B Technical Library Washington, DC 20360
2	Commander Naval Ship Engineering Center ATTN: Technical Library NSEC 6105G Hyattsville, MD 20782
1	Commander David W. Taylor Naval Ship Research & Development Center ATTN: L42-3 Library Bethesda, MD 20084
1	Commander Naval Surface Weapons Center ATTN: DX-21, Library Br. Dahlgren, VA 22448
3	Commander Naval Surface Weapons Center ATTN: Code WA501/Navy Nuclear Programs Office Code WX21/Tech Lib Code 240/C. J. Aronson Silver Spring, MD 20910
1	Commander Naval Weapons Center ATTN: Code 533, Tech Lib China Lake, CA 93555
2	Commander Naval Ship Research and Development Center Facility Underwater Explosions Research Division ATTN: Code 17, W.W. Murray Technical Library Portsmouth, VA 23709
2	Commander Naval Weapons Evaluation Facility ATTN: Document Control R. Hughes Kirtland AFB Albuquerque, NM 87117

DISTRIBUTION LIST

No. of Copies	Organization
4	Officer-in-Charge Civil Engineering Laboratory Naval Construction Btn Center ATTN: Stan Takahashi R. J. Odello John Crawford Technical Library Port Hueneme, CA 93041
2	Commander Naval Research Laboratory ATTN: Code 2027, Tech Lib Code 8440, F.Rosenthal Washington, DC 20375
1	Superintendent Naval Postgraduate School ATTN: Code 2124, Tech Rpts Lib Monterey, CA 93940
1	HQ USAF (IN) Washington, DC 20330
1	HQ USAF (PRE) Washington, DC 20330
2	AFSC (DLCAW; Tech Lib) Andrews AFB Washington, DC 20331
2	AFATL (ATRD/ R. Brandt) Eglin AFB, FL 32542
2	ADTC (ADBRL-2; Tech Lib) Eglin AFB, FL 32542
2	RADC (EMTLD/Docu Lib; EMREC, R.W. Mair) Griffiss AFB, NY 13340
1	AFWL/SUL Kirtland AFB, NM 87117
1	AFWL/DE-I Kirtland AFB, NM 87117

No. of Copies	Organization
1	AFWL/DEX Kirtland AFB, NM 87115
1	AFWL/Robert Port Kirtland AFB, NM 87117
1	AFWL/DEV, Jimmie L. Bratton Kirtland AFB, NM 87117
1	AFWL/R. Henny Kirtland AFB, NM 87117
1	AFWL/DEV, M.A. Plamondon Kirtland AFB, NM 87117
2	Commander-in-Chief Stragegic Air Command ATTN: NRI-STINFO Lib XPFS Offut AFB, NB 68113
1	AFML (MAMD/Dr. T. Nicholas) Wright-Patterson AFB, OH 45433
4	FTD (TDFBD; TDPMG; ETET, CAPT R.C. Husemann; TD-BTA/Lib) Wright-Patterson AFB, OH 45433
1	ASD (Tech Lib) Wright-Patterson AFB, OH 45433
2	ADC (XP; XPQDQ) Wright-Patterson AFB, OH 45433
1	AFIT (Lib Bldg. 640, Area B) Wright-Patterson AFB, OH 45433
1	Director US Bureau of Mines ATTN: Technical Library Denver Federal Center Denver, CO 80225

DISTRIBUTION LIST

No. of Copies	Organization
1	Director US Bureau of Mines Twin Cities Research Center ATTN: Technical Library P. O. Box 1660 Minneapolis, MN 55111
1	US Energy Research and Development Administration Division of Headquarters Svcs ATTN: Doc Control for Classified Tech Library Library Branch G-043 Washington, DC 20545
1	US Energy Research and Development Administration Albuquerque Operations Office ATTN: Doc Control for Tech Lib P. O. Box 5400 Albuquerque, NM 87117
1	US Energy Research and Development Administration Nevada Operations Office ATTN: Doc Control for Tech Lib P. O. Box 14100 Las Vegas, NV 89114
5	Director Lawrence Livermore Laboratory ATTN: L.W. Woodruft/L-96 Tech Info Dept L-3 D. M. Norris/L-90 Ted Butkovich/L-200 J. R. Hearst/L-205 P. O. Box 808 Livermore, CA 94550
4	Director Lawrence Livermore Laboratory ATTN: Jack Kahn/L-7 J. Carothers/L-7 Robert Schock/L-437 R. G. Dong/L-90 P. O. Box 808 Livermore, CA 94550
4	Director Los Alamos Scientific Laboratory ATTN: Doc Control for Rpts Lib R. A. Gentry G. R. Spillman Al Davis P. O. Box 1663 Los Alamos, NM 87544
1	Director National Aeronautics and Space Administration Scientific and Technical Information Facility P. O. Box 8575 Baltimore/Washington International Airport, MD 21240
3	Aerospace Corporation ATTN: Tech Info Services (2 cys) P. N. Mathur P. O. Box 92957 Los Angeles, CA 90009
1	Agbabian Associates ATTN: M. Agbabian 250 North Nash Street El Segundo, CA 90245
1	Analytic Services, Inc. ATTN: George Hesselracher 5613 Leesburg Pike Falls Church, VA 22041
1	Applied Theory, Inc. ATTN: John G. Trulio 1010 Westwood Boulevard Los Angeles, CA 90024
1	Artec Associates, Inc. ATTN: Steven Gill 26046 Eden Landing Road Hayward, CA 94545
1	AVCO ATTN: Res Lib A830, Rm 7201 201 Lowell Street Wilmington, MA 01887

DISTRIBUTION LIST

No. of Copies	Organization
1	The BDM Corporation ATTN: Richard Hensley P. O. Box 9274 Albuquerque International Albuquerque, NM 87119
2	The Boeing Company ATTN: Aerospace Library R. H. Carlson P. O. Box 3707 Seattle, WA 98124
1	Brown Engineering Co., Inc. ATTN: Manu Patel Cummings Research Park Huntsville, AL 35807
2	California Research and Technology, Inc. ATTN: Ken Kreyenhagen Technical Library 6269 Variel Avenue Woodland Hills, CA 91364
1	Calspan Corporation ATTN: Technical Library P. O. Box 235 Buffalo, NY 14221
1	Civil/Nuclear Systems Corporation ATTN: Robert Crawford 1200 University N. E. Albuquerque, NM 87102
1	EG&G, Incorporated Albuquerque Division ATTN: Technical Library P. O. Box 10218 Albuquerque, NM 87114
1	The Franklin Institute ATTN: Zemons Zudans 20th Street and Parkway Philadelphia, PA 19103
1	General American Trans Corporation General American Research Division ATTN: G. L. Neidhardt 7449 N. Natchez Avenue Niles, IL 60648
1	General Electric Company-TEMPO ATTN: DASIAC P. O. Drawer QQ Santa Barbara, CA 93102
1	General Research Corporation ATTN: Benjamin Alexander P. O. Box 3587 Santa Barbarba, CA 93105
2	Hazeltine Corp ATTN: Carl Meinen Greenlawn, NY 11740
1	J. H. Wiggins Co., Inc. ATTN: John Collins 1650 South Pacific Cost Highway Redondo Beach, CA 90277
6	Kaman Avidyne ATTN: Dr. N. P. Hobbs (4 cys) Mr. S. Criscione Mr. John Calligeros 83 Second Avenue Northwest Industrial Park Burlington, MA 01830
3	Kaman Sciences Corporation ATTN: Library P. A. Ellis F. H. Shelton 1500 Garden of the Gods Road Colorado Springs, CO 80907

DISTRIBUTION LIST

No. of Copies	Organization
1	Lockheed Missiles & Space Co. ATTN: Technical Library P. O. Box 504 Sunnyvale, CA 94088
2	Martin Marietta Aerospace Orlando Division ATTN: G. Fotieo Mail Point 505, Craig Luongo P. O. Box 5837 Orlando, FL 32805
3	McDonnell Douglas Astronautics Corporation ATTN: Robert W. Halprin Mr. C. Gardiner Dr. P. Lewis 5301 Balsa Avenue Huntington Beach, CA 92647
2	Merritt Cases, Inc. ATTN: J. L. Merritt Technical Library P. O. Box 1206 Redlands, CA 92373
1	Meteorology Research, Inc. ATTN: W. D. Green 464 West Woodbury Road Altadena, CA 91001
1	The Mitre Corporation ATTN: Library P. O. Box 208 Bedford, MA 01730
2	Pacifica Technology ATTN: G. Kent R. Bjork P. O. Box 148 Del Mar, CA 92014
4	Physics International Corp. ATTN: E. T. Moore Dennis Orphal Coye Vincent F. M. Sauer 2700 Merced Street San Leandro, CA 94577
4	Physics International Corp ATTN: Robert Swift Charles Godfrey Larry A. Behrmann Technical Library 2700 Merced Street San Leandro, CA 94577
5	R&D Associates ATTN: Dr. H. L. Brode Dr. Albert L. Latter C. P. Knowles William B. Wright Henry Cooper P. O. Box 9695 Marina del Rey, CA 90291
4	R&D Associates ATTN: Jerry Carpenter Sheldon Schuster J. G. Lewis Technical Library P. O. Box 9695 Marina del Rey, CA 90291
1	The Rand Corporation ATTN: C. C. Mow 1700 Main Street Santa Monica, CA 90406

DISTRIBUTION LIST

No. of Copies	Organization
6	Sandia Laboratories ATTN: Doc Control for 3141 Sandia Rpt Collection A. J. Chaban M. L. Merritt L. J. Vortman W. Roherty L. Hill Albuquerque, NM 87115
1	Sandia Laboratories Livermore Laboratory ATTN: Doc Control for Tech Lib P. O. Box 969 Livermore, CA 94550
1	Science Applications, Inc. ATTN: Technical Library P. O. Box 3507 Albuquerque, NM 87110
2	Science Applications, Inc. ATTN: R. Seebaugh John Mansfield 1651 Old Meadow Road McLean, VA 22101
1	Science Applications, Inc. 8201 Capwell Drive Oakland, CA 94621
2	Science Applications, Inc. ATTN: Technical Library Michael McKay P. O. Box 2351 La Jolla, CA 92038
4	Systems, Science & Software ATTN: Donald R. Grine Ted Cherry Thomas D. Riney Technical Library P. O. Box 1620 La Jolla, CA 92037
3	Terra Tek, Inc. ATTN: Sidney Green Technical Library A. H. Jones 420 Wakara Way Salt Lake City, UT 84108
2	Tetra Tech, Inc. ATTN: Li-San Hwang Technical Library 630 North Rosemead Blvd. Pasadena, CA 91107
7	TRW Systems Group ATTN: Paul Lieberman Benjamin Sussholtz Norm Lipner William Rowan Jack Farrell Pravin Bhutta Tech Info Ctr/S-1930 One Space Park Redondo Beach, CA 90278
1	TRW Systems Group ATTN: Greg Hulcher San Bernardino Operations P. O. Box 1310 San Bernardino, CA 92402
2	Union Carbide Corporation Holifield National Laboratory ATTN: Doc Control for Tech Lib Civil Defense Research Proj P. O. Box X Oak Ridge, TN 37830
1	Universal Analytics, Inc. ATTN: E. I. Field 7740 W. Manchester Blvd. Playa del Rey, CA 90291

DISTRIBUTION LIST

No. of Copies	Organization
1	Weidlinger Assoc. Consulting Engineers ATTN: M. L. Baron 110 East 59th Street New York, NY 10022
1	Westinghouse Electric Company Marine Division ATTN: W. A. Votz Hendy Avenue Sunnyvale, CA 94008
2	Battelle Memorial Institute ATTN: Technical Library R. W. Klingesmith 505 King Avenue Columbus, OH 43201
1	California Institute of Technology ATTN: T. J. Ahrens 1201 E. California Blvd. Pasadena, CA 91109
2	COSMIC ATTN: L. C. Gadol 112 Barrow Hall University of Georgia Athens, GA 30602
2	Denver Research Institute University of Denver ATTN: Mr. J. Wisotski Technical Library P. O. Box 10127 Denver, CO 80210
3	IIT Research Institute ATTN: Milton R. Johnson R. E. Welch Technical Library 10 West 35th Street Chicago, IL 60616
2	Lovelace Foundation for Medical Education ATTN: Asst. Dir. of Research/ Robert K. Jones Technical Library 5200 Gibson Blvd, SE Albuquerque, NM 87108
1	Massachusetts Institute of Technology Aeroelastic and Structures Research Laboratory ATTN: Dr. E. A. Witmer Cambridge, MA 02139
2	Southwest Research Institute ATTN: Dr. W. E. Baker A. B. Wenzel 8500 Culebra Road San Antonio, TX 78206
2	Stanford Research Institute ATTN: Dr. G. R. Abrahamson Carl Peterson 333 Ravenswood Avenue Menlo Park, CA 94025
1	University of Dayton Industrial Security Super. KL-505 ATTN: H. F. Swift 300 College Park Avenue Dayton, OH 45409
1	University of Illinois Consulting Engineering Services ATTN: Nathan M. Newmark 1211 Civil Engineering Building Urbana, IL 61801

DISTRIBUTION LIST

No. of Copies	Organization
2	The University of New Mexico The Eric H. Wang Civil Engineering Rsch Facility ATTN: Larry Bickle Neal Baum University Station Box 188 Albuquerque, NM 87131
2	Washington State University Administration Office ATTN: Arthur Miles Hohorf George Duval Pullman, WA 99163

Aberdeen Proving Ground

Marine Corps Ln Ofc
Dir, USAMSAA
ATTN: Dr. J. Sperrazza
Mr. R. Norman, GWD
B. E. Cummings
M. Reches
Cmdt, USAOC&S
ATTN: ATSL-CD-CD
LT C. E. Tate